64'er

3. SONDERHEFT FÜR C 128-FANS

SONDERHEFT 22 ÖS 100,-/Sfr. 14,- Lit. 12000/hfl. 18,-/dkr. 72,- DM 14,-

Markt&Technik

128er



Viele Super-Programme zum Abtippen

Weltrekord im Kopieren

★ 8-Sekunden-Copy für zwei Laufwerke

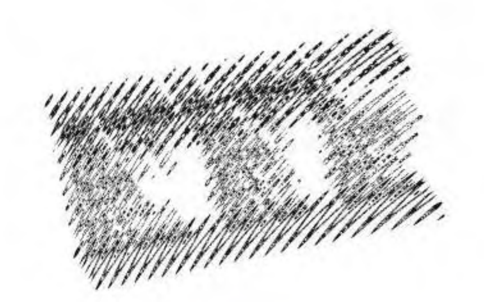
Assembler der Spitzenklasse

★ Neu: Zugriff auf beide Prozessoren
★ Unglaublich: Z80- und 8502-Codes mischbar

Nützliche Tricks und interessante Anwendungen

★ Farbiges Super-Scrolling
★ Grafik im 80-Zeichen-Modus
★ Der C 128 ordnet Ihr Video-Archiv
★ Fesselndes Action-Spiel

Alle Programme auch auf Diskette erhältlich







64ER ONLINE



Aller guten Dinge...



...sind drei. Und dies in doppeltem Sinne. Zum einen halten Sie das dritte Sonderheft zum C128 in Händen, zum anderen soll auf die drei Herzen Ihres Computers hingewiesen werden. C128, C64, CP/M - diese drei Computertypen sind beim Commodore 128 in einem Gehäuse vereint.

Für den C64-Modus gibt es nahezu alles an Software. Bei der eigentlichen Stärke dieses Computers, dem C128-Modus, sieht es allerdings nicht so rosig aus. Darum haben wir in dieser Ausgabe viele interessante Listings zusammengestellt. Dieses C128-Sonderheft soll Ihnen helfen, die Fähigkeiten ihres Computers nicht nur besser zu nutzen sondern auch wesentlich zu erweitern.

So war es beispielsweise bislang nicht möglich, die komfortablen Grafik-Befehle des Basic 7.0 im 80-Zeichen-Modus zu verwenden. Das Programm »Color-Pack« bietet sogar mehr als dies. Eine Auflösung von 640 mal 200 Punkten und Schriften in allen Größen und Richtungen seien als zwei Eigenschaften von vielen genannt.

Die Basic-Erweiterung »Master« richtet unter anderem im C128 eine RAM-Disk mit bis zu 58 KByte Größe ein. Hier können Sie beispielsweise Programme, Grafik-Bilder oder Funktionstasten-Belegungen hinterlegen und blitzschnell darauf zugreifen.

»X-Basic« ergänzt das Basic 7.0 um etliche hilfreiche Befehle. Nur ein Beispiel: Auf eine Umwandlungs-Routine für Hardcopies, die Multicolor-Grafiken in Graustufen wiedergibt, haben Sie vielleicht schon lange gewartet. Zusätzlich wird erklärt, wie Sie eigene Basic-Befehle in die Interpreter-Routine des Computers einbinden können.

Faszinierende Grafiken erzeugt der »MSG« (Modul-Screen-Generator). Mit einem Trick lassen sich bis zu 999 dieser Grafiken im Speicher unterbringen und in eigene Basic-Programme einbinden. Der Grafikschwerpunkt in dieser Ausgabe enthält noch weitere interessante Listings wie »Apfelberge«, eine Variante der bekannten Apfelmännchen oder einen Scroll-Generator für farbige Laufschriften.

Nicht nur Basic-Programmierer, sondern auch Assembler-Fans kommen in diesem Sonderheft auf ihre Kosten. Ein Leckerbissen ist der Assembler »Double-Ass«. Beide Prozessoren - der 8502 und der Z80 - lassen sich damit komfortabel programmieren. Die Prozessor-Codes sind sogar innerhalb eines Programms mischbar. Sie können Maschinensprache-Routinen, die der Z80 erheblich schneller als sein Kollege erledigt, zukünftig diesem Prozessor überlassen. Übrigens, mit »Master« können Sie Z80-Routinen auch aus Basic-Programmen aufrufen.

Da Double-Ass hardwaremäßig in den Computer integriert werden kann, finden Sie in den Grundlagen wichtige Informationen über den »freien Steckplatz« - im alten und neuen C128. Eine kurze Einführung in die Z80-Programmierung ist ebenfalls enthalten.

Durch den 80-Zeichen-Modus ist der C128 ausgezeichnet für professionelle Anwendungen geeignet. »Double-Touch« ist eine davon. Wenn Sie dieses Listing abgetippt haben und zwei 1571-Laufwerke besitzen, sparen Sie zukünftig beim Diskettenkopieren viel Zeit. Eine komplette Diskette wird in 8 Sekunden kopiert. Eine Geschwindigkeit, die auch Besitzer eines Personal Computers vor Neid erblassen läßt.

Wenn Sie ein Freund von Videofilmen sind, können Sie ab jetzt das Archivieren Ihrer Kassetten dem C128 überlassen. Doch nicht nur Kassetten müssen archiviert werden. Jeder Computerbesitzer wird früher oder später die Übersicht verlieren, wenn seine Programme nicht sinnvoll geordnet sind. Der »Disk-Manager« hilft dabei enorm. Durch individuelle Rubriken und viele Suchfunktionen behalten Sie immer den Überblick. Für die wissenschaftlich orientierten Leser ist auch eine interessante Anwendung dabei. Wenn Sie demnächst Experimente auswerten und einen mathematischen Zusammenhang suchen, können Sie diese mühsame Arbeit Ihrem Computer überlassen. Der C128 sucht selbständig die richtige Funktion zu Ihren Versuchsergebnissen.

Viele Tips und Tricks bringen den Programmierern unter Ihnen neue Impulse und Arbeitshilfen. Sie erfahren, in welchen ungewöhnlichen Varianten man den Boot-Sektor nutzen kann, wie man den Fast-Modus beim 40-Zeichen-Bildschirm ohne Bildverlust verwendet, welche Befehle einen Interlace-Modus wie beim Amiga generieren, und... und... und...

Natürlich muß man sich auch einmal entspannen. Was ist besser dazu geeignet als ein gutes Computerspiel? Nur gibt es nicht viele davon für den C128-Modus. Ein Test informiert Sie über zwei sehr interessante kommerzielle Spiele. Zwei Spiele, die Sie sicher begeistern werden, können Sie sogar abtippen. Also - hinein ins Vergnügen.

Ihr Gottfried Knechtel (Redakteur)


PROGRAMM-
SERVICE

INHALT

Grafik

Tips & Tricks

Programmierhilfen

Grundlagen

Anwendung

Spiele

Spiele-Listings

Bücher

64'er-Referenz

Eingabehilfen

Sonstiges

Alle Programme aus Artikeln mit dem ■ -Symbol finden Sie auch auf der Programmservice-Diskette zu diesem Sonderheft (siehe linke Seite)

Soft-Scrolling in Farbe

Gute Spiele sind häufig durch scrollende Laufschriften optisch aufgewertet. Eine kurze Maschinenroutine ermöglicht jedem C128-Anwender, farbige Laufschriften im 80-Zeichen-Modus auf einfache Weise zu erstellen.

Durch Soft-Scrolling oder das punktweise Verschieben des Bildschirms lassen sich interessante Effekte erzielen. Mit diesem Programm können Sie auf einfache Weise farbige Laufschriften wie in Bild 1 erzeugen. Da Soft-Scrolling im 80-Zeichen-Modus nicht leicht zu realisieren ist, bedient man sich hier eines Tricks: Laufschrift durch Umdefinierung des Zeichensatzes.

Laufschrift mit Trick

Damit Sie bald in den Genuß des Soft-Scrollings kommen, ohne mit programmtechnischen Details geplagt zu werden, sollte zunächst Listing 1 mit dem MSE im C64-Modus und Listing 2 mit »Checksummer 128« (Seite 158) eingegeben werden. »SCROLLMASCHINE« (bitte den Filenamen beim MSE richtig angeben!) ist eine kurze Maschinenroutine. Sie wird mit »BLOAD"SCROLLMASCHINE",ON B0« geladen und mit folgender Syntax aufgerufen:

SYS DEC ("0B00"),LÄNGE,MODUS,TEMPO

Dabei steht LÄNGE für die Anzahl der Zeichen, die die Laufschrift umfaßt. Die Informationen für die Laufschrift selbst holt sich die Routine aus dem Speicherbereich 3584 bis 3839 ($0e00 bis $0eff); die Länge wird in Adresse 3583 ($0dff) vom Programm gespeichert.

Der Parameter MODUS legt fest, mit welchem Zeichensatz die Laufschrift dargestellt werden soll (208 = Groß-/Grafik-Schrift; 216 = Klein-/Groß-Schrift).

TEMPO bestimmt die Scroll-Geschwindigkeit und kann Werte von 0 (extrem schnell) bis 255 (äußerst langsam) annehmen.

Programmierung der Maschinenroutine

Die Position, an der die Laufschrift erscheint, wird jedoch nicht durch derartige Parameter eingestellt. Stattdessen müssen Sie folgendermaßen vorgehen:

1. Lassen Sie alle Texte, außer der Laufschrift, im Klein-/Groß-Zeichensatz des C128 ausgeben.



Bild 1. Ein Beispiel für farbige Laufschrift. In Wirklichkeit scrollt die kleine Schrift hinter dem Schriftzug »64'er« durch.

2. Schalten Sie dann auf Groß-/Grafik-Schrift und drucken eine, mit »B« beginnende und alphabetisch geordnete Zeichenfolge wie

BCDEFGHIJKLMNOPQRSTUVWXYZ

3. Starten Sie die Maschinenroutine mit den beschriebenen Parametern. An allen Stellen, an denen eine solche Zeichenfolge steht, rollt die Laufschrift ab. Dabei dürfen

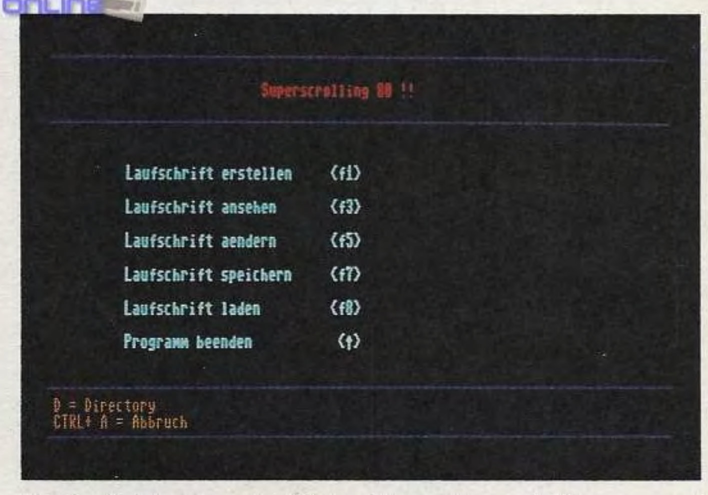


Bild 2. Auf komfortable Weise können Laufschriften erzeugt werden. Hier das Hauptmenü.

```
Name : scrollmaschine     0b00 0c80
-----------------------------------
0b00 : 8d 53 0b 8d 0a 0c 8e 24  2f
0b08 : 0c 8c 28 0b a9 10 a0 20  a4
0b10 : 8d 44 0b 8c 49 0b 20 43  08
0b18 : 0b 20 09 0c 20 33 0b 20  ff
0b20 : b8 0b 20 dd 02 4c 4e 0b  f3
0b28 : 01 10 20 00 00 00 00 00  3a
0b30 : 00 00 00 20 e4 0b ad 29  e4
0b38 : 0b 8d 44 0b ad 2a 0b 8d  f0
0b40 : 49 0b 60 a9 00 8d 29 0b  83
0b48 : a9 00 8d 2a 0b 60 20 43  55
0b50 : 0b 78 a9 1a 85 fd a9 00  34
0b58 : 85 a6 20 c3 0b 20 c3 0b  88
0b60 : 20 97 0b 20 d2 0b 20 02  1d
0b68 : 0c a5 a6 c9 08 d0 eb 20  21
0b70 : e4 0b 20 09 91 d0 08 c6  50
0b78 : fd d0 db 58 4c 4e 0b 58  f3
0b80 : 85 ff 60 a2 13 20 cc cd  72
0b88 : 98 ca 20 cc cd 60 ee 29  15
0b90 : 0b d0 03 ee 2a 0b 60 18  4f
0b98 : 2a 08 48 a4 a6 b9 2b 0b  68
0ba0 : f0 04 68 09 01 48 68 aa  17
0ba8 : a9 00 28 90 02 a9 01 99  12
0bb0 : 2b 0b 8a 85 a7 e6 a6 60  c1
0bb8 : a0 07 a9 00 99 2b 0b 88  76
0bc0 : 10 fa 60 ad 29 0b ac 2a  0d
0bc8 : 0b 20 83 0b 20 d8 cd 85  31
0bd0 : a7 60 ad 29 0b ac 2a 0b  0d
0bd8 : 20 83 0b a5 a7 20 ca cd  74
0be0 : 20 8e 0b 60 ad 29 0b 38  d7
0be8 : e9 18 8d 29 0b b0 03 ce  46
0bf0 : 2a 0b 60 ad 29 0b 18 69  8c
0bf8 : 08 8d 29 0b 90 03 ee 2a  a4
0c00 : 0b 60 ae 28 0b ca d0 fd  32
0c08 : 60 a9 1a 85 fc a0 00 84  52
0c10 : fe a4 fe b9 ff 0d c9 60  a7
0c18 : 90 03 29 7f 2c 29 3f 85  78
0c20 : f0 a9 00 a2 d8 85 fa 86  ec
0c28 : fb a4 f0 18 a5 fa 69 08  9c
0c30 : 85 fa 90 02 e6 fb 88 d0  a9
0c38 : f2 a0 00 a2 01 a9 fa 8d  33
0c40 : aa 02 20 a2 02 99 2b 0b  f7
0c48 : c8 c0 08 d0 ee 20 dd 02  f8
0c50 : a0 00 84 ff b9 2b 0b 85  3d
0c58 : a7 20 d2 0b e6 ff a4 ff  26
0c60 : c0 08 d0 f0 20 f3 0b e6  12
0c68 : fe a4 fc c4 fe b0 a2 20  d0
0c70 : f3 0b 60 15 15 15 15 15  1c
0c78 : 15 15 15 15 15 15 15 15  78
```

Listing 1. »SCROLLMASCHINE« bitte mit dem MSE (Seite 158) im C64-Modus eingeben

derartige Zeichenfolgen durchaus an mehreren Bildschirmpositionen gleichzeitig vorhanden sein; auch die Länge ist variabel, BCDEF gibt beispielsweise nur ein fünf Zeichen umfassendes »Fenster« für die Laufschrift an.

4. Durch Löschen aller in Groß-/Grafik-Schrift geschriebenen Zeichen und Setzen der Laufschrift-Länge auf Null wird die Laufschrift ausgeblendet.

Laufschrift-Editor – zugleich ein Demoprogramm

»LAUFSCHR.-EDITOR« (Listing 2) ermöglicht Ihnen das Testen von Laufschriften und ist aufgrund der Menüsteuerung (Bild 2) problemlos zu bedienen. Von diesem Basic-Programm wird die Maschinenroutine »SCROLLMASCHINE« (Listing 1) automatisch nachgeladen.

Das Programm speichert die im Bereich 3584 bis 3589 befindliche Laufschrift auch auf Diskette; sie wird über BLOAD"NAME",ON B0 eingelesen.

Dank der zahlreichen Kommentare, die sich in REM-Anweisungen befinden, ist der Laufschrift-Editor zusätzlich ein lehrreiches Beispiel für die Programmierung der Scroll-Routine, mit der Sie sicherlich noch recht viele Ideen realisieren können!

(Nino Cimerman/F. Müller)

Hinweis: Die Grundlagen zur VDC-Programmierung im allgemeinen und zur Zeichensatzdefinition im besonderen wurden in der Artikelserie »VDC – 80 Zeichen ist nicht alles« in den 64'er-Stammheften 3/87, 5/87, 7/87 und 8/87 ausführlich besprochen.
Der Quelltext zu »SCROLLMASCHINE« befindet sich im Hypra-Ass-Format auf der Leserservice-Diskette zu diesem Sonderheft unter dem Filenamen »SCROLL-MASCH.SRC«.

```
10 REM ************************************     <A99>
15 REM *                                  *     <ATB>
20 REM *  SUPER-SCROLLING 80 FUER PC 128  *     <AHA>
30 REM *              V1.0                *     <APB>
40 REM *                                  *     <B1C>
60 REM ************************************     <BHE>
70 :                                            <GQ3>
80 TRAP 6140                                    <VQ6>
81 :                                            <N2S>
85 REM --- D8 = KLEINSCHRIFT-MODUS ---          <8L2>
86 :                                            <MQR>
90 Z$="D8"                                      <MA1>
99 :                                            <LIM>
100 BLOAD "SCROLLMASCHINE": REM --- MASCHINENPROG
    RAMM LADEN   ---                            <PQK>
110 KEY 2,CHR$(138)      : REM --- FUNKTIONSTASTE
    N FUER MENUE ---                            <DHA>
120 KEY 1,CHR$(133)      : REM --- VORBEREITEN.
               ---                              <BDT>
140 KEY 3,CHR$(134)                             <MMM>
160 KEY 5,CHR$(135)                             <LQN>
180 KEY 7,CHR$(136)                             <KUN>
190 KEY 8,CHR$(137)                             <[illegible]>
200 GS=1                 : REM GS = GESCHWINDIKEI
    T DER LAUFSCHRIFT                           <A52>
210 FAST                                        <DHJ>
220 :                                           <0C1>
500 REM ========   HAUPTMENUE ========          <GI2>
501 :                                           <640>
510 TT$="---------------------------------------
    ": TT$=TT$+TT$                              <2QJ>
520 PRINT CHR$(14): SCNCLR : PRINT "{2HOME,CLR,LI
    G.BLUE}"TT$                                 <UKM>
540 PRINT "{DOWN}" TAB(27) CHR$(14) CHR$(15)"{LIG
    .RED}SUPERSCROLLING 80 !!" CHR$(143)        <E7R>
560 PRINT "{DOWN,LIG.BLUE}"TT$                  <FB3>
580 PRINT TAB(8) CHR$(159)"{2DOWN,2SPACE}LAUFSCHR
    IFT ERSTELLEN{5SPACE}<F1>"                  <94F>
600 PRINT TAB(8)"{DOWN,2SPACE}LAUFSCHRIFT ANSEHEN
    {7SPACE}<F3>"                               <8SA>
620 PRINT TAB(8)"{DOWN,2SPACE}LAUFSCHRIFT AENDERN
    {7SPACE}<F5>"                               <EG3>
630 PRINT TAB(8)"{DOWN,2SPACE}LAUFSCHRIFT SPEICHE
    RN{5SPACE}<F7>"                             <9DG>
640 PRINT TAB(8)"{DOWN,2SPACE}LAUFSCHRIFT LADEN{9
    SPACE}<F8>"                                 <JS6>
650 PRINT TAB(8)"{DOWN,2SPACE}PROGRAMM BEENDEN{11
    SPACE}<↑>"                                  <5VI>
660 PRINT "{2DOWN,LIG.BLUE}"TT$                 <V0D>
680 PRINT "{2DOWN,LIG.BLUE}"TT$                 <FCC>
690 CHAR 1,1,21,"{BROWN}D = DIRECTORY"          <LNE>
695 CHAR 1,1,22,"CTRL+ A = ABBRUCH"             <LDF>
700 :                                           <4CH>
710 REM --- BEFEHLS-TASTEN AUSWERTUNG  ----     <HQ7>
711 :                                           <4KI>
720 GET KEY X$: IF X$="↑" THEN 6160             <RCV>
725 IF X$="D" THEN BEGIN                        <ACK>
726 : PRINT "{ORANGE}";: WINDOW 50,6,79,19,1    <IHC>
727 : DIRECTORY "?????????????.SC"              <KHL>
730 : PRINT "{LIG.GREEN}BITTE EINE TASTE DRUECKEN
    .{LIG.BLUE}";: GET KEY X$ : PRINT "{CLR,2HOME
    }"                                          <KB3>
732 BEND : GOTO 720                             <OMD>
734 IF X$="{CTRL+A}" THEN PRINT CHR$(159);: END <LII>
735 X=ASC(X$)-132                               <I3U>
740 ON X GOTO 1000,2000,3000,4000,5000          <SV4>
760 GOTO 720                                    <PU8>
800 :                                           <BDD>
1000 REM  ==== LAUFSCHRIFT ERSTELLEN ====       <34C>
1001 :                                          <184>
1002 REM --- DER STEHENDE TEXT WIRD IM KLEINSCHRI
     FT-MODUS GESCHRIEBEN  ---                  <3CD>
1003 :                                          <106>
1005 PRINT "{CLR,LIG.BLUE}" CHR$(14)TT$: Z=0    <FDL>
1007 PRINT TAB(27)"{DOWN,LIG.RED}LAUFSCHRIFT ERST
     ELLEN": PRINT "{DOWN,LIG.BLUE}"TT$         <PBH>
1008 CHAR ,25,19,"NICHT MEHR ALS 64 ZEICHEN !!" <557>
1009 CHAR 1,0,18,"{LIG.BLUE}"+TT$: CHAR 1,0,21,TT
     $: PRINT "{HOME,6DOWN}"                    <6SV>
1010 INPUT "{ORANGE}TEXT : {RVSON}";TX$: GOSUB 10
     20: GOTO 520                               <801>
1011 :                                          <106>
1012 REM --- HIER WIRD DIE LAUFSCHRIFT IN DAS RAM
      GEPOKET ---                               <34C>
1013 :                                          <184>
1020 FOR I=0 TO 255: POKE DEC("0E00")+I,32: NEXT
     : REM BEREICH ERST LOESCHEN                <JHV>
1029 :                                          <288>
1030 FOR I=1 TO LEN (TX$)                       <CM0>
1040 : X$=MID$(TX$,I,1)                         <NR6>
1060 : X=ASC(X$)                                <L7A>
1080 : BANK 0: POKE DEC("0DFF")+I,X             <QGP>
1100 NEXT                                       <A90>
1110 POKE DEC("0DFF"),LEN(TX$): REM LAENGE DER LA
     UFSCHRIFT NACH $0DFF                       <TM4>
1120 RETURN                                     <8K7>
1999 :                                          <EPQ>
2000 REM   ====  LAUFSCHRIFT ANSEHEN  ====      <040>
2020 :                                          <60P>
2040 Z=0: SCNCLR                                <Q1H>
2060 PRINT TAB(5)"{LIG.BLUE}LAUFSCHRIFT ..."    <3SS>
2080 PRINT TAB(5)"SCHNELLER{11SPACE}= CURSOR RECH
     TS                                         <A0M>
2100 PRINT TAB(5)"LANGSAMER{11SPACE}= CURSOR LINK
     S                                          <AUD>
2120 PRINT TAB(5)"GROSS-SCHRIFT{7SPACE}= <G>    <P28>
2140 PRINT TAB(5)"GROSS/KLEIN-SCHRIFT = <K>     <E1G>
2160 PRINT TAB(5)"ZURUECK ZUM MENUE{3SPACE}= <↑> <UG1>
2180 :                                          <209>
2200 CHAR 1,4,12,"{LIG.RED}SUPER-{2SPACE}"+CHR$(1
     42)+"BCDEF"+CHR$(14)+"{LIG.RED,2SPACE}-SCROL
     LING 80 !"                                 <FMQ>
2220 PRINT CHR$(142);                           <IU7>
2240 :                                          <70T>
2260 FOR B=1 TO 3: COLOR 5,(B+4): FOR I=7 TO 12:
     X=X+1: CHAR ,30+X,I,"BCDEF": NEXT I,B      <CQS>
2279 :                                          <4OI>
2280 X=2: FOR B=1 TO 3: COLOR 5,(B+4)           <OBI>
2281 :                                          <184>
2300 FOR I=17 TO 1 STEP -1: X=X+1: CHAR ,30+X,I,"
     BCDEF": NEXT I,B                           <6S2>
2301 :                                          <48G>
2305 REM --- WO TZ$ IM GROSS-SCHRIFT-MODUS STEHT
     WIRD GESCROLLT ---                         <104>
2306 :                                          <5GN>
2320 TZ$="BCDEFGHIJKLMNOPQRSTUVWXYZ[£]↑← !"     <JSM>
2340 TZ$=TZ$+CHR$(34)+"#$%&'()*+,-./0123456789:;<
     =>?*"                                      <HFQ>
2360 A=LEN(TX$): IF A<1 THEN A=PEEK(3583) : REM "
     LAENGE DER LAUFSCHRIFT SETZEN"             <LKH>
2380 TY$=LEFT$(TZ$,A-1): CHAR ,10,24,"{ORANGE}"+T
     Y$                                         <NSJ>
2400 SW=10                                      <1CN>
2420 PRINT "{HOME}";CHR$(14)                    <Q05>
2440 CHAR ,70,0,"": PRINT USING "###";GS        <L2P>
2460 SYS DEC("0B00"),A,DEC(Z$),GS               <7PU>
2476 :                                          <105>
2477 REM --- DIE LETZTE GEDRUECKTE DASTE WIRD VON
      DER MASCHINEN-ROUTINE ---                 <30C>
2478 REM --- IN SPEICHERSTELLE 255 UEBERGEBEN.
                                ---             <2SB>
2479 :                                          <20A>
2480 X=PEEK(255)                                <BKN>
2481 :                                          <78S>
2500 IF X=DEC("9D") THEN 2620 : REM " ↑-TASTE   <JSA>
2520 IF X=DEC("1D") THEN 2700 : REM " CURSOR RECH
     TS{3SPACE}(TEMPO STEIGERN)                 <30A>
2540 IF X=DEC("5E") THEN 2800 : REM CURSOR LINKS
        (TEMPO VERRINGERN)                      <5LM>
2560 IF X=DEC("4B") THEN Z$="D8": GOTO 2440: REM
     K- BZW. G-INSTR ASTE (RCLR ODUS)           <T6E>
```

Listing 2. »LAUFSCHR.-EDITOR« gestattet das Editieren/Austesten von Laufschriften und wird mit »Checksummer 128« eingegeben



```
2580 IF X=DEC("47") THEN Z$="D0": GOTO 2440: REM
     " --''--{2SPACE}--''--                          <CDM>
2600 GOTO 2440                                       <10K>
2619 :                                               <30E>
2620 REM GESCHWINDIGKEIT ERHOEHEN                    <3KE>
2621 :                                               <080>
2640 IF GS =<1 THEN GS=1: GOTO 2440                  <7VM>
2660 GS=GS-SW                                        <DP6>
2680 GOTO 2440                                       <30K>
2699 :                                               <70U>
2700 REM GESCHWINDIGKEIT DROSSELN                    <3KE>
2701 :                                               <080>
2720 IF GS=>255 THEN GS=255: GOTO 2440               <FCS>
2740 GS=GS+SW                                        <EPA>
2760 GOTO 2440                                       <0OK>
2780 :                                               <40H>
2800 GOTO 520: REM  ZURUECK ZUM MENUE                <2G5>
2820 :                                               <E1P>
3000 REM ====  LAUFSCHRIFT AENDERN  ====             <144>
3020 :                                               <40H>
3040 PRINT "{CLR,LIG.BLUE}" CHR$(14)TT$              <97C>
3060 PRINT TAB(27)"{DOWN,LIG.RED}LAUFSCHRIFT AEND
     ERN ": PRINT "{DOWN,LIG.BLUE}"TT$               <3AU>
3080 CHAR 1,0,18,"{LIG.BLUE}"+TT$: CHAR 1,0,21,TT
     $: PRINT "{HOME,6DOWN}"                         <7OR>
3100 X$=""                                           <IUP>
3101 :                                               <48G>
3105 REM --- LAUFSCHRIFT AUS RAM-BANK 0 AUSLESEN
     ---                                             <104>
3120 :                                               <50L>
3140 FOR I=1 TO PEEK(DEC("0DFF"))                    <4BU>
3160 : BANK 0: X=PEEK(DEC("0DFF")+I)                 <M8K>
3180 : X$=X$+CHR$(X)                                 <MKU>
3200 NEXT                                            <IE0>
3220 :                                               <60P>
3240 PRINT "{ORANGE}TEXT : {RVSON,2RIGHT}";X$;"{H
     OME,7DOWN,7RIGHT}";: TX$="": INPUT TX$          <LA8>
3260 GOSUB 1020                                      <NNF>
3280 GOTO 520                                        <1UC>
3300 :                                               <60P>
4000 REM ==== LAUFSCHRIFT ABSPEICHERN ====           <64O>
4020 :                                               <A19>
4040 NA$="": FI$="{26SPACE}"                         <A3A>
4060 PRINT "{CLR,LIG.BLUE}"TT$: PRINT "{DOWN}" TA
     B(27)"{LIG.RED}LAUFSCHRIFT ABSPEICHERN"         <LO9>
4080 PRINT "{DOWN,LIG.BLUE}"TT$                      <S3E>
4100 CHAR 1,0,18,"{BROWN}MAXIMAL 13 ZEICHEN !{LIG
     .BLUE}"                                         <ENL>
4120 INPUT "{HOME,6DOWN,GREY1}NAME : ";NA$           <SVM>
4140 IF LEN (NA$) >13 THEN 4040                      <MIG>
4160 IF NA$="" THEN 520                              <HCG>
4180 :                                               <E1P>
4200 NA$=NA$+RIGHT$(FI$,13-LEN(NA$))                 <D68>
4220 BSAVE (NA$+".SC"),D0,P 3583 TO P 3840   : RE
     M IN $0DFF-$0EFF LIEGT DER TEXT                 <LE3>
4240 GOSUB 6080: GOTO 520                            <EE1>
4259 :                                               <9P6>
5000 REM ==== LAUFSCHRIFT LADEN  ====                <74S>
5020 :                                               <811>
5040 NA$="": FI$="{26SPACE}"                         <AJA>
5060 PRINT "{CLR,LIG.BLUE}"TT$: PRINT "{DOWN}" TA
     B(27)"{LIG.RED}LAUFSCHRIFT LADEN"               <ER0>
5080 PRINT "{DOWN,LIG.BLUE}"TT$                      <C3E>
5100 CHAR 1,0,18,"{BROWN}MAXIMAL 13 ZEICHEN !{LIG
     .BLUE}"                                         <6ML>
5120 INPUT "{HOME,6DOWN,GREY1}NAME : ";NA$           <OVM>
5140 IF NA$="" THEN 520                              <HKJ>
5160 IF LEN (NA$) >13 THEN 5040                      <7KK>
5180 :                                               <C1H>
5200 NA$=NA$+RIGHT$(FI$,13-LEN(NA$))                 <DMA>
5220 BLOAD (NA$+".SC") : GOSUB 6080                  <FCR>
5240 GOTO 520                                        <3EC>
5260 :                                               <811>
6000 REM ====== FEHLERBEHANDLUNGSROUTINEN ======     <44G>
6020 :                                               <E1P>
6040 REM ---- FUER FLOPPY ----                       <54K>
6060 :                                               <C1H>
6080 CHAR ,5,24,DS$ : GET KEY A$ : PRINT "{HOME}"
     : RETURN                                        <AT0>
6100 :                                               <E1P>
6120 REM ---- RUN/STOP-SCHUTZ ----                   <44G>
6140 RESUME NEXT                                     <QUQ>
6160 RUN "TAKE OFF ..." : REM -- MENUE-PROGRAMM L
     ADEN  --                                        <QUQ>
```

Listing 2. »LAUFSCHR.-EDITOR« gestattet das Editieren und Testen von Laufschriften (Schluß)

Hochauflösende Grafik im 80-Zeichen-Modus des C 128

Jetzt können auch Basic-Programmierer den VDC-Chip vollkommen ausnutzen – eine neue Basic-Erweiterung läßt die normalen Grafikbefehle des C 128 auch im 80-Zeichen-Modus wirksam werden, wobei eine Grafikauflösung von 480 mal 240 Punkten erreicht wird.

Die komfortablen Grafikbefehle des C 128 gestatten die Erstellung von hochauflösenden Grafiken auf sehr bequeme Weise. Doch gilt dies nur für den 40-Zeichen-Bildschirm, der eine Grafikauflösung von immerhin 320 mal 200 Einzelpunkten zuläßt. Der zweiten Bildschirm-Modus des C 128, die 80-Zeichen-Darstellung, wird von diesen Anweisungen allerdings nicht berührt, was um so unverständlicher ist, wenn man erfährt, daß sich hier Grafikbilder der doppelten Auflösung von 640 mal 200 entwerfen lassen.

Aus diesem Grund führen selbst einfache Grafikaktionen wie das Setzen und Löschen von Punkten in diesem Grafik-Modus zu einer mühsamen Auseinandersetzung mit dem VDC-Chip. Denn jener für die fantastische 80-Zeichen-Grafik verantwortliche Baustein ist kompliziert zu programmieren, wobei man ohne Assembler-Kentnisse kaum auskommt.

Besitzer der Basic-Erweiterung »Color-Pack 1« kennen jedoch solche Probleme nicht, denn nach dem Booten dieses Programmes stehen alle bekannten Grafik-Befehle wie

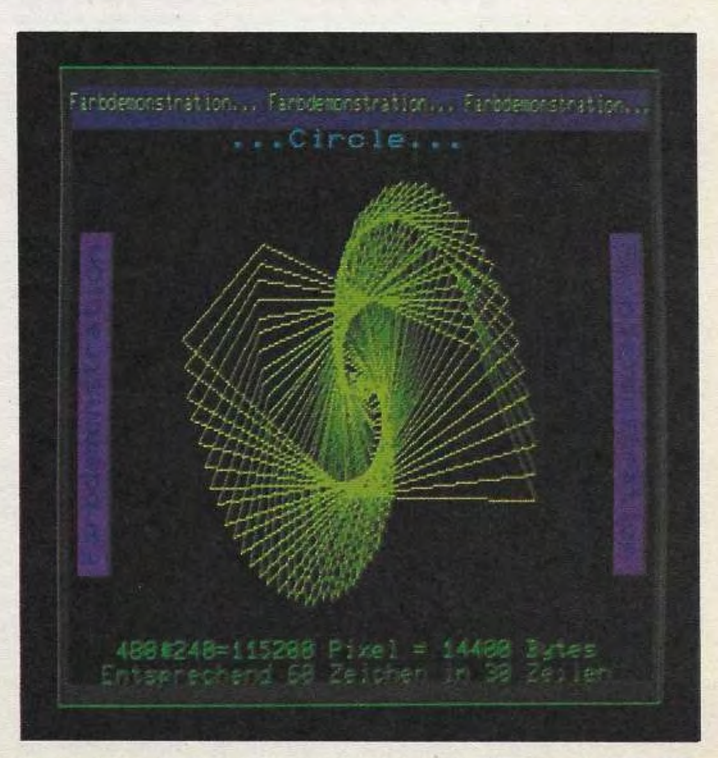


Bild 1. Eine eindrucksvolle Grafik im 80-Zeichen-Modus, die sich mit »Color-Pack 1« schnell erstellen läßt.

GRAPHIC, COLOR, CHAR, DRAW oder CIRCLE bereit, um solch eindrucksvolle Grafiken wie in Bild 1 auf den 80-Zeichen-Monitor zu zaubern. Doch Color-Pack 1 ist nicht nur »irgendeine« Grafik-Erweiterung. Einige Befehle wurden in ihrer Funktion modifiziert und verwöhnen den Anwender mit noch größerer Leistungsfähigkeit.

Die CHAR-Anweisung zur Darstellung von Text innerhalb einer Grafik dreht nun einen Text in jedem beliebigen Winkel, während zwei neue Parameter die Ausdehnung der Schrift in Höhe und Breite bestimmen. Schmal- oder Kleinschrift in jeder nur erdenklichen Richtung ist damit problemlos zu verwirklichen (Bild 2).

Mit dem SCALE-Befehl läßt sich endlich der Koordinatenbereich beliebig setzen, da das Color-Pack auch mit Gleitkommazahlen arbeitet. So darf die X-Achse zum Beispiel von $-2*\pi$ bis $3*\pi$ skaliert werden. Wer mathematische Funktionen auf dem Bildschirm darstellen will, wird diese Verbesserung zu schätzen wissen.

COLOR packt die Farben an

Eine besonders starke Wandlung mit angenehmen Folgen erfuhr der COLOR-Befehl des C 128. Mit seiner Hilfe lassen sich ähnlich dem Multicolor-Modus des 40-Zeichen-Bildschirms bis zu drei Farben für Vorder- und Hintergrund wählen, auf die die Grafikbefehle später schnell und einfach zugreifen können. Ein neuer COLOR-Modus gibt dem Anwender zusätzlich die Fähigkeit, Teile seiner Grafik nachträglich einzufärben, ohne dabei Gefahr zu laufen, ein bereits erstelltes Bild zu zerstören.

Zwei weitere Befehle des Basic 7.0 erhielten vollkommen neue Funktionen. GSHAPE und SSHAPE dienen nun zur Sicherung der Grafik auf Diskette oder in einen der beiden Bildbereiche, die jeweils eine 80-Zeichen-Grafik aufnehmen können. Eine modifizerte Belegung der Funktionstasten (Tabelle 1) erlaubt das Umschalten beider Grafikbereiche auf Knopfdruck.

Ungewöhnliches Format

Darüber hinaus präsentiert sich der Bildschirm mit Color-Pack 1 in einer neuen und zunächst ungewöhnlichen Form. Statt der üblichen Auflösung von 640 mal 200 Punkten erhält der Computer-Grafiker eine Arbeitsfläche von 480 mal 240 Grafikpunkten. Damit stehen anstelle von 128000 Pixels nurmehr 115200 ansprechbare Punkte zur Verfügung. Dieser scheinbare Nachteil erweist sich jedoch bald als recht vorteilhaft, da sich hierbei ein fast quadratisches Bild ergibt, das eine ausreichende Auflösung in Y-Richtung bietet. Der enorme Auflösungsunterschied von 640 zu 200 Punkten wird damit zum Großteil ausgeglichen. Daneben entspricht das neue Format in X-Richtung genau der erreichbaren Auflösung eines Epson- oder kompatiblen Druckers für DIN-A4-Formate, so daß ein Ausdruck ohne Verzerrungen erfolgen kann.

Selbstverständlich besitzt das Color-Pack für diesen Zweck eine Hardcopy-Routine, die den Ausdruck der Grafik auf einem Epson-Drucker und Kompatiblen in verschiedenen Größen gestattet. Bild 3 zeigt eine Hardcopy der Grafik in Bild 1. Der Druck erfolgt jedoch nur als Schwarzweiß-Bild. Unterschiedliche Farben werden dabei nicht (als Graustufen) berücksichtigt.

Bevor jedoch das Color-Pack 1 seine Leistungsfähigkeit demonstrieren kann, sind zunächst die Programme »color-pack 1.m« sowie »color-pack.key.m« mit dem MSE im C 64-Modus einzugeben und zu speichern. Sie finden diese in Listing 1 und 2. Listing 3 stellt schließlich den Lader



Bild 2. Eine nicht alltägliche Besonderheit: Schriften lassen sich in Größe und Lage beliebig verändern

von Color-Pack 1 dar. Er sollte mit dem Checksummer 128, den Sie in dieser Ausgabe finden, eingetippt werden. Der Lader wird unter dem Namen »color-pack 1« auf der Diskette gespeichert und später mit RUN gestartet. Der Computer sollte sich dabei natürlich im 80-Zeichen-Modus befinden und Ihr RGB-Monitor eingeschaltet sein. Außerdem müssen Sie vor dem Start des Laders die ASCII/DIN-Taste drücken, so daß der Computer auf den deutschen Zeichensatz geschaltet wird.

Color-Pack 1 installiert sich nun im Speicher des C 128 und steht dann mit erstaunlichen Grafikbefehlen bereit, die wir im folgenden Abschnitt näher betrachten wollen. Wir werden hier allerdings nur jene Anweisungen erläutern, die einer Funktionsänderung unterworfen sind. Die verbleibenden Befehle werden wie bei der Arbeit mit der

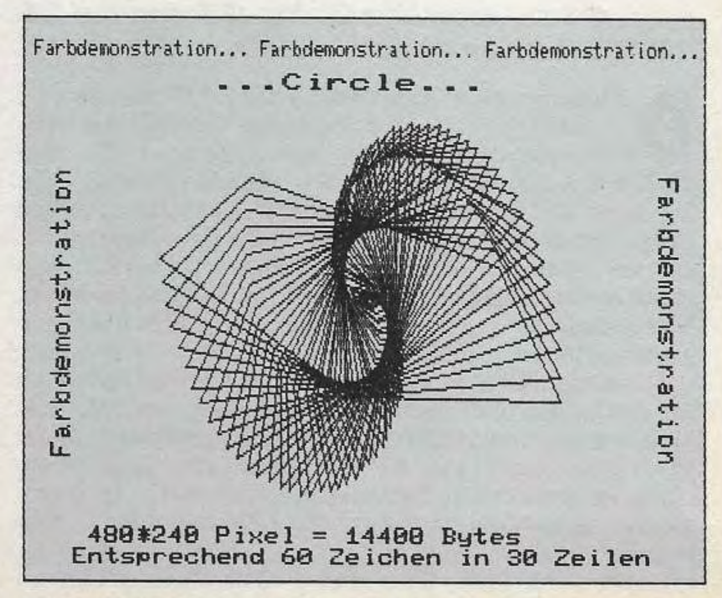


Bild 3. Diese Hardcopy von Bild 1 kann in zwei unterschiedlichen Formaten gedruckt werden.

40-Zeichen-Grafik verwendet. Die genaue Funktion können Sie dem Handbuch zum C128 entnehmen.

Funktionsänderungen mit Folgen

GRAPHIC1, <Vordergrundfarbe>, <Hintergrundfarbe>
schaltet in den 80-Zeichen-Grafikmodus. Dabei dürfen die Werte (0-15) der Zeichen- und die Hintergrundfarbe des gesamten Bildschirms angegeben werden. Fallen diese Angaben weg, werden die voreingestellten Farbwerte übernommen. Die Grafik-Modi 2 bis 4 sind nicht mehr verwendbar und erzeugen einen »ILLEGAL QUANTITY ERROR«.

GRAPHIC 0 oder **GRAPHIC**
desaktiviert den Grafikmodus. Der C128 kehrt in die Textdarstellung zurück.

GRAPHIC Modus, Rand
druckt eine Hardcopy der aktuellen Grafik auf einem Epson-Drucker und Kompatiblen. »Modus« bestimmt dabei die Größe des Druckbildes, während »Rand« die Wahl des Abstandes zum linken Papierrand erlaubt. Mögliche Kombinationen sind:

Befehl	Wirkung	Bereich von »Rand«
GRAPHIC 10, Rand	kleines Bild	0 bis 68
GRAPHIC 11,0	doppelte Bildbreite (genaue Blattbreite DIN A4)	
GRAPHIC 12, Rand	doppelte Bildhöhe	0 bis 68
GRAPHIC 13,0	großes Bild mit doppelter Bildbreite und -höhe	

Beispiel:
```
GRAPHIC 10,34
```
Es wird ein kleines Bild genau in der Blattmitte gedruckt.

COLOR 1, Vordergrundfarbe, Hintergrundfarbe
COLOR 2, Vordergrundfarbe, Hintergrundfarbe
COLOR 3, Vordergrundfarbe, Hintergrundfarbe

Mit COLOR lassen sich drei Farbquellen für Vordergrund und Hintergrund unabhängig setzen, die in den Zeichenbefehlen durch Angabe der Farbquell-Nummer (1 bis 3) verwendet werden können.
Beispiele:
```
BOX 2,10,10,100,100
```
Es wird ein Rechteck mit der Vordergrund- und Hintergrundfarbe aus Farbquelle 2 gezeichnet.
```
CIRCLE 3,120,100,50,25
```
Es wird ein Kreis mit den Farben aus Farbquelle 3 gezeichnet.

Die Angabe der Farbquelle darf auch weggelassen werden. In diesem Fall entspricht die Zeichen- und Hintergrundfarbe den im Farb-RAM befindlichen Farbwerten. Die Ausführung eines Befehls wird dadurch erheblich beschleunigt:
```
CIRCLE,120,100,50,25
```

COLOR 7, Vordergrundfarbe, Hintergrundfarbe, x1, y1, x2,y2
färbt das durch (x1,y1) und (x2,y2) vorgegebene Rechteck in der angegebenen Vorder- und Hintergrundfarbe, ohne die Grafik zu zerstören. Dabei stehen die Koordinaten (x1,y1) für die linke obere (x2,y2) für die rechte untere Ecke des Rechtecks.

Die Farben können nicht pixelweise, sondern nur in Grafikblöcken zu je 8 mal 8 Punkten gesetzt werden, woraus sich für x1, x2, y1 und y2 folgende Bereichsgrenzen ergeben:
$0 <= x1 <= x2 <= 59$
$0 <= y1 <= y2 <= 29$

CHAR Farbquelle,x, y, Winkel, a$, x-Breite, x-Höhe
schreibt einen beliebigen Text (a$) mit der Farbe der gewünschten Farbquelle in den Grafikbildschirm an Position (x,y), und dreht ihn im angegebenen Winkel (0 bis 360 Grad). Auch hier führt ein Weglassen der Farbquelle zu einer deutlichen Beschleunigung der Ausführung. Da die »schrägen« Schriften aber nur schwer lesbar sind, empfiehlt es sich, nur die Winkel 0, 90, 180 und 270 Grad zu verwenden.

Die letzten beiden Parameter bestimmen die Breite und Höhe der Buchstaben. Sie dürfen Werte von 0 bis 255 annehmen und sind bei den Schreibrichtungen von 0 und 180 Grad von Bedeutung.

Bei Winkeln von 90 und 270 Grad, also bei senkrechter Schreibrichtung, sind zwei weitere Parameter wichtig:

CHAR Farbquelle, x, y, Winkel, a$, y-Breite, y-Höhe
»y-Breite« und »y-Höhe« geben nun die Breite und Höhe der Buchstaben an. »x-Breite« und »x-Höhe« sind hier ohne Wirkung und können mit leeren Kommata übergangen werden. Ein kleines Beispiel in Listing 4 demonstriert die sich daraus ergebende Schriftvielfalt (Bild 2).

Beliebige Skalierung des Grafikbereichs

SCALE 3, xf, xs, yf, ys
ändert die Skalierung des Grafikbildschirms. Damit sind nun beliebige Definitionsbereiche in X- und Y-Richtung möglich. Die Parameter stellen jedoch nicht die jeweiligen Unter- und Obergrenzen für die X- und Y-Achse dar, sondern bestimmen die Skalierungsfaktoren und -summanden der Koordinatenachsen. Will man festgelegte Grenzen definieren, sind deshalb einige Berechnungen nötig:
```
xmin,xmax   : Unter- und Obergrenze der X-Achse
ymin,ymax   : Unter- und Obergrenze der Y-Achse
xf= (479-0)/(xmax-xmin)
xs= 0-xf*xmin
yf= (0-239)/(ymax-ymin)
ys= 239-yf*ymin
SCALE 3,xf,xs,yf,ys
```
Ein Beispiel finden Sie in Listing 5, das mit veränderter Skalierung eine Sinuskurve erstellt.

Die veränderte Skalierung wirkt nun auf die Befehle LOCATE und DRAW, wobei auch Fließkommawerte gestattet sind. Haben Sie beispielsweise die X- und Y-Achse jeweils von -10 bis 10 definiert, lassen sich nun mit DRAW Punkte oder Linien zeichnen, wie etwa:
```
DRAW 1,-5,-5
DRAW 2,-10,-5 TO -3.45,4.76
```
Die restlichen Grafik-Befehle beziehen sich jedoch nicht auf die veränderte Skalierung. Sie werden weiterhin im normalen Koordinatennetz gezeichnet.

LOCATE x,y
setzt den Grafikcursor im definierten Bereich an beliebige Koordinaten, wobei auf die Skalierung mit SCALE 3 Rücksicht genommen wird. Ein Beispiel für die oben definierten Bereiche ist:
```
LOCATE -2.65,4.72
```
Von dem mit LOCATE angesteuerten Punkt aus kann dann normal gekennzeichnet werden.

LOCATE x, y, Variable
Mit dieser Variation von LOCATE kann getestet werden, ob an Koordinate (x,y) ein Punkt gesetzt ist oder nicht. Bei einem gesetzten Punkt erhält die angegebene Variable den Wert 1, andernfalls bekommt sie den Wert 0 zugewiesen.

SSHAPE Bildbereich
sichert das aktuelle Grafikbild in einem der beiden Bildbereiche (1 oder 2). So lassen sich zwei Grafikbilder gleichzeitig im Speicher des C128 ablegen und bearbeiten.

SSHAPE Bildbereich, "Bildname"
speichert den angegebenen Bildbereich (1 oder 2) unter einem beliebigen Namen auf Diskette.

Grafiken sichern

GSHAPE Bildbereich
kopiert, als Pendant zu SSHAPE, die im angegebenen Bildbereich abgelegte Grafik zur weiteren Bearbeitung zurück in den aktuellen Grafikbildschirm.

GSHAPE Bildbereich, "Bildname"
lädt eine Grafik des 80-Zeichen-Bildschirms in einen der beiden Bildbereiche (1 oder 2).

Taste	Belegung
F1	löscht Bildschirm unterhalb des Cursors
F2	DLOAD
F3	DIRECTORY
F4	DSAVE
F5	Umschalten der beiden Grafikbilder
F6	Listing auf Drucker ausgeben
F7	LIST
F8	MONITOR

Tabelle 2. Color-Pack 1 belegt die Funktionstasten neu

Damit wären alle Besonderheiten von Color-Pack 1 erläutert. Die verbleibenden Grafik-Anweisungen CIRCLE, BOX und PAINT sind wie gewohnt verwendbar, arbeiten nun aber wie alle anderen Grafik-Befehle im 80-Zeichen-Grafikmodus. Wurde das Color-Pack gestartet, ist die 40-Zeichen-Grafik nicht mehr aktivierbar. Man wird sie allerdings nicht vermissen.

(Rainer Franzen/Michael Thomas)

```
Name : color-pack 1.m     2000 2d3f
-----------------------------------
2000 : 20 cf 23 a2 04 20 52 9e  ed
2008 : 20 f2 9d 20 1c 9e e0 02  4b
2010 : 90 03 4c 28 7d 8a 4a 6a  64
2018 : 85 8b 10 04 a5 83 f0 07  30
2020 : 20 cc 23 b0 02 d0 01 60  f1
2028 : 20 ea 92 8d 03 ff a5 33  41
2030 : 85 24 a5 34 85 25 38 a5  65
2038 : 35 e9 03 85 1b a5 36 e9  5f
2040 : 00 85 1c a2 00 86 63 86  2d
2048 : 64 ae 33 11 d0 03 ce 34  bb
2050 : 11 ce 33 11 20 cc 23 b0  0e
2058 : 02 d0 ee ee 33 11 d0 03  61
2060 : ee 34 11 20 c9 23 ae 31  84
2068 : 11 d0 03 ce 32 11 ce 31  c5
2070 : 11 a5 63 20 d4 20 85 63  5c
2078 : 18 ad 31 11 69 02 8d 31  15
2080 : 11 90 03 ee 32 11 a5 64  83
2088 : 20 d4 20 85 64 ae 31 11  6e
2090 : d0 03 ce 32 11 ce 31 11  4a
2098 : ee 33 11 d0 03 ee 34 11  19
20a0 : 20 cc 23 b0 02 d0 bc a2  e4
20a8 : 03 a0 00 a5 25 c5 34 d0  a3
20b0 : 06 a5 24 c5 33 f0 1a a5  b9
20b8 : 24 d0 02 c6 25 c6 24 20  f7
20c0 : b7 03 8d 03 ff 9d 31 11  90
20c8 : ca 10 ec 20 b5 4b 4c 43  47
20d0 : 20 4c f2 9d 48 20 cc 23  86
20d8 : b0 18 f0 16 68 d0 16 aa  4e
20e0 : a8 a5 25 c5 1c 90 0f d0  81
20e8 : 06 a5 24 c5 1b 90 07 4c  6e
20f0 : 3a 4d 68 a9 00 60 bd 31  7c
20f8 : 11 8d 04 ff 91 24 8d 03  47
2100 : ff e6 24 d0 02 e6 25 e8  53
2108 : e0 04 d0 ea a9 80 60 20  dc
2110 : cf 23 a2 1f 20 6d 9e a2  2b
2118 : 2b 20 52 9e 20 06 9e 8c  82
2120 : 54 11 8d 55 11 20 1c 9e  cb
2128 : e0 02 90 03 4c 28 7d 8e  a7
2130 : 6c 11 8a 48 20 89 63 68  7d
2138 : d0 1c f0 03 20 0b 64 20  df
2140 : e0 22 ad 4e 11 d0 f5 a2  1b
2148 : 04 bd 5b 11 9d 30 11 ca  59
2150 : d0 f7 8e 6c 11 60 a2 00  ec
2158 : ad 49 11 4a 90 02 a2 02  df
2160 : bd 60 11 8d 5a 11 bd 61  2b
2168 : 11 8d 5b 11 a9 00 a2 03  64
2170 : 9d 56 11 ca 10 fa a2 07  47
2178 : bd 31 11 48 ca 10 f9 20  71
2180 : e0 22 a2 00 68 9d 31 11  74
2188 : e8 e0 08 d0 f7 ad 5a 11  75
2190 : d0 05 ce 5b 11 30 b0 ce  f5
2198 : 5a 11 a2 25 a0 1b ad 49  f4
21a0 : 11 4a 90 02 a0 19 a9 00  b4
21a8 : 4a 48 20 6d 9d 9d 31 11  7a
21b0 : 98 9d 32 11 68 90 02 09  eb
21b8 : a0 e8 e8 a0 19 4e 49 11  66
21c0 : 90 02 a0 1b 2e 49 11 e0  10
21c8 : 27 f0 dd a2 06 0a f0 bd  23
21d0 : 90 08 fe 31 11 d0 03 fe  ec
21d8 : 32 11 0a ca ca 10 f1 30  c4
21e0 : 95 ea ea ea a2 01 86 83  56
21e8 : 20 86 03 c9 a4 f0 0b 20  84
21f0 : cf 23 20 86 03 d0 03 4c  85
21f8 : ab 23 20 86 03 c9 2c f0  1f

2200 : 05 c9 a4 f0 01 60 48 20  a5
2208 : 80 03 a2 04 20 70 9e 68  04
2210 : 10 06 20 e0 22 4c fa 21  fa
2218 : 20 f2 9d 20 ab 23 4c fa  18
2220 : 21 20 cf 23 a2 1f 20 52  f2
2228 : 9e 20 06 9e 8c 54 11 8d  f6
2230 : 55 11 20 06 9e 8c 56 11  a0
2238 : 8d 57 11 08 a2 23 20 4a  0f
2240 : 9d 28 b0 11 ad 54 11 8d  1c
2248 : 56 11 ad 55 11 24 d8 10  f3
2250 : 04 0e 56 11 2a 8d 57 11  a2
2258 : 20 06 9e 8c 5c 11 8d 5d  f4
2260 : 11 20 06 9e 8c 5e 11 8d  f2
2268 : 5f 11 20 06 9e 85 77 98  3e
2270 : a4 77 20 77 9a a2 2d a0  7c
2278 : 2b 20 7c 9d 90 0e c9 68  77
2280 : a0 01 20 70 9d 9d 31 11  65
2288 : 98 9d 32 11 a2 03 bd 54  7f
2290 : 11 9d 58 11 ca 10 f7 a9  08
2298 : 90 20 f3 9a a2 07 bd 54  8a
22a0 : 11 9d 60 11 ca 10 f7 20  07
22a8 : 50 67 20 f2 9d a2 02 20  49
22b0 : 1e 9e 8a d0 03 4c 28 7d  08
22b8 : 8e 20 12 18 ad 20 12 6d  dd
22c0 : 5c 11 8d 5c 11 90 03 ee  13
22c8 : 5d 11 a2 2d a0 2b 20 7c  d9
22d0 : 9d b0 08 20 50 67 20 e0  4e
22d8 : 22 90 e1 a0 2d 20 52 67  bb
22e0 : a2 02 a0 06 a9 00 9d 3d  f8
22e8 : 11 9d 3e 11 20 99 9d 10  df
22f0 : 08 de 3d 11 de 3e 11 d0  9f
22f8 : 0b c9 00 d0 04 c0 00 f0  2a
2300 : 03 fe 3d 11 9d 39 11 0a  f0
2308 : 9d 41 11 98 9d 3a 11 2a  e1
2310 : 9d 42 11 ca ca a0 04 e0  ef
2318 : 00 f0 c9 a2 0a a0 08 20  5d
2320 : 7c 9d a9 00 2a 2a 8d 47  8e
2328 : 11 49 02 8d 48 11 18 a9  d1
2330 : 10 6d 47 11 a8 48 49 02  e1
2338 : aa 20 7c 9d 9d 31 11 98  9e
2340 : 9d 32 11 68 a8 18 a9 08  4a
2348 : 6d 48 11 aa 20 7c 9d 8d  ea
2350 : 45 11 8c 46 11 20 ab 23  11
2358 : ac 48 11 38 b9 39 11 e9  f1
2360 : 01 99 39 11 b0 0b b9 3a  5d
2368 : 11 e9 00 99 3a 11 b0 01  92
2370 : 60 ae 47 11 ad 46 11 30  cd
2378 : 06 20 9a 23 ae 48 11 18  3b
2380 : ad 45 11 7d 41 11 8d 45  21
2388 : 11 ad 46 11 7d 42 11 8d  6d
2390 : 46 11 ae 48 11 20 9a 23  d6
2398 : f0 bb a0 02 18 bd 31 11  25
23a0 : 7d 3d 11 9d 31 11 e8 88  04
23a8 : d0 f3 60 ad 6c 11 0d 6b  9a
23b0 : 11 f0 16 ee 31 11 d0 03  82
23b8 : ee 32 11 20 c9 23 ae 31  db
23c0 : 11 d0 03 ce 32 11 ce 31  1d
23c8 : 11 4c b2 26 4c d0 26 4c  ed
23d0 : 96 26 78 a9 47 8d 08 03  d4
23d8 : a9 24 8d 09 03 a9 28 8d  51
23e0 : 02 03 a9 24 8d 03 03 58  00
23e8 : 20 45 a8 a0 00 a2 07 8e  37
23f0 : 06 d5 84 2f 84 31 84 33  32
23f8 : a9 85 85 30 85 32 85 34  34
2400 : a9 ea 85 9e 84 b0 a9 2c  20
2408 : 85 9f a9 44 85 b1 8d 02  70

2410 : ff b1 9e 91 b0 c8 d0 f9  4a
2418 : 8d 03 ff a9 2d 85 84 a9  c0
2420 : 25 85 85 a9 09 85 86 60  36
2428 : a2 ff 86 3c 20 93 4f 86  dc
2430 : 3d 84 3e 20 80 03 aa f0  f0
2438 : ef b0 03 4c e2 4d 20 0a  f7
2440 : 43 20 86 03 4c 4a 24 20  7d
2448 : 80 03 c9 de 90 1d c9 ea  87
2450 : b0 19 aa bd 93 23 8d 63  3e
2458 : 24 bd 9f 23 8d 64 24 20  74
2460 : 80 03 20 89 24 20 86 03  fe
2468 : 4c f6 4a 20 86 03 4c f3  60
2470 : 4a 89 00 f4 0f 21 f6 f8  e5
2478 : 43 7d 0f 79 d0 24 20 26  68
2480 : 21 22 29 29 29 2a 25 6a  6f
2488 : 28 f0 3e 20 f4 87 e0 01  cd
2490 : f0 04 90 35 b0 76 a2 01  98
2498 : 20 21 25 a9 e0 a0 24 20  ab
24a0 : f4 24 a9 00 a2 12 20 cc  e6
24a8 : cd e8 20 cc cd a2 1f 8e  16
24b0 : 00 d6 a2 40 a0 38 20 bf  98
24b8 : 24 a5 84 a2 c0 a0 07 20  92
24c0 : cf cd ca d0 fa 88 10 f7  67
24c8 : 60 a9 ea a0 24 20 f4 24  2b
24d0 : 20 0c ce 4c de 6a 01 02  7d
24d8 : 04 06 07 14 15 19 22 23  0d
24e0 : 3c 5c 28 1e 24 38 40 c7  ad
24e8 : 74 5a 50 66 27 19 20 08  36
24f0 : 00 47 7d 64 20 45 a8 8d  6a
24f8 : 03 25 8c 04 25 a0 09 be  2a
2500 : d6 24 b9 ea 24 20 cc cd  c6
2508 : 88 10 f4 60 4c a5 2a 20  bc
2510 : f4 87 e0 07 f0 2e e0 01  e7
2518 : 90 04 e0 04 90 03 4c e5  81
2520 : 69 86 83 b5 83 29 0f aa  77
2528 : 20 d2 25 85 9b a6 83 b5  14
2530 : 83 4a 4a 4a 4a aa 20 d2  d4
2538 : 25 0a 0a 0a 0a 05 9b a6  ab
2540 : 83 95 83 60 a6 83 b5 83  df
2548 : 85 87 8a 48 a2 04 20 21  4a
2550 : 25 68 85 83 20 bb 25 85  fb
2558 : b0 20 be 25 85 b1 20 bb  4a
2560 : 25 ea ea 38 e5 b0 90 67  b1
2568 : 85 b4 20 be 25 ea ea 38  ed
2570 : e5 b1 90 5b 85 b5 a9 40  ea
2578 : 85 9e a9 38 85 9f a5 b0  0b
2580 : 20 b1 25 a9 00 85 bd a5  66
2588 : bd c5 b1 f0 07 20 af 25  2d
2590 : e6 bd d0 f3 20 45 a8 a5  21
2598 : b4 85 b6 20 61 26 a5 87  ad
25a0 : 20 ca cd c6 b6 10 f7 20  7d
25a8 : af 25 c6 b5 10 e9 60 a9  77
25b0 : 3c 18 65 9e 85 9e 90 02  b9
25b8 : e6 9f 60 a9 3c 2c a9 1e  c3
25c0 : 8d ca 25 20 5c 79 20 f4  fc
25c8 : 87 e0 00 b0 02 8a 60 20  0c
25d0 : 28 7d 20 1e 9e b0 02 8a  0f
```

Listing 1. »color-pack 1.m« enthält die wichtigen Grafik-Routinen. Bitte mit dem MSE (Seite 158) im C64-Modus eingeben



```
25d8 : 60 e0 11 b0 f2 8a f0 ef  2a
25e0 : bd 4b 6a 60 ad 00 ff 85  cf
25e8 : 9b 20 45 a8 ad 34 11 d0  5c
25f0 : 61 ac 33 11 c0 f0 b0 5a  a1
25f8 : ad 31 11 ae 32 11 85 a3  61
2600 : 86 a4 f0 07 ca d0 4b c9  e9
2608 : e0 b0 47 46 a4 29 07 85  96
2610 : a4 66 a3 46 a3 46 a3 38  04
2618 : 90 1f a6 83 f0 1b b5 83  18
2620 : 48 98 4a 4a 4a 20 6e 26  3c
2628 : 18 69 40 85 9e 98 69 38  7a
2630 : 85 9f 20 61 26 68 20 ca  75
2638 : cd ad 33 11 20 6e 26 85  e4
2640 : 9e 84 9f 20 61 26 20 d8  86
2648 : cd 85 a5 a6 a4 bd 59 26  00
2650 : 18 60 38 90 01 60 4c 28  4d
2658 : 7d 80 40 20 10 08 04 02  7f
2660 : 01 a5 9f a2 12 20 cc cd  61
2668 : a5 9e e8 4c cc cd 85 9e  ae
2670 : 8d 88 26 a9 00 85 9f a2  f0
2678 : 06 06 9e 26 9f ca d0 f9  75
2680 : a2 04 a5 9e a4 9f 38 e9  5d
2688 : c8 b0 01 88 ca d0 f7 18  3d
2690 : 65 a3 90 01 c8 60 a2 01  27
2698 : c9 2c d0 04 a9 18 d0 09  dd
26a0 : 20 f4 87 e0 04 b0 08 a9  71
26a8 : 38 8d 17 26 86 83 60 4c  d0
26b0 : 28 7d 20 e4 25 b0 13 a6  ad
26b8 : 83 d0 05 49 ff 25 a5 2c  26
26c0 : 05 a5 48 20 61 26 68 20  d7
26c8 : ca cd a5 9b 8d 00 ff 60  ef
26d0 : ad 17 26 85 be a9 18 8d  f8
26d8 : 17 26 20 e4 25 a6 be 8e  47
26e0 : 17 26 b0 e6 25 a5 f0 06  63
26e8 : 20 ca 26 a2 00 60 20 ca  64
26f0 : 26 a2 ff 60 20 c8 28 30  bd
26f8 : 03 4c d7 67 20 86 03 20  87
2700 : 96 26 a2 1f 20 52 9e 20  85
2708 : 06 9e 8c 5c 11 8d 5d 11  21
2710 : a9 00 a2 07 9d 60 11 ca  fa
2718 : 10 fa 85 b1 20 5c 79 20  48
2720 : 7b 87 85 b0 20 1e 9e 90  65
2728 : 03 8e 94 28 20 1e 9e 90  2b
2730 : 03 8e 7f 28 20 1e 9e 90  ee
2738 : 03 8e 8d 28 20 1e 9e 90  79
2740 : 03 8e 86 28 a4 b1 c4 b0  7d
2748 : f0 28 8d 02 ff b1 24 a2  53
2750 : 00 8e 00 ff 20 5b 27 e6  df
2758 : b1 d0 e9 c9 ff 90 02 69  84
2760 : 7e a8 c9 12 d0 04 a9 04  c3
2768 : d0 06 c9 92 d0 05 a9 00  dc
2770 : 85 f3 60 c9 91 d0 0f 38  8c
2778 : ad 62 11 e9 04 8d 62 11  30
2780 : b0 03 ce 63 11 60 c9 11  2f
2788 : d0 0f 18 ad 62 11 69 04  f8
2790 : 8d 62 11 90 03 ee 63 11  fc
2798 : 60 29 7f c9 20 90 f9 98  45
27a0 : c9 80 b0 0c c9 60 90 04  41
27a8 : 29 df b0 08 29 bf 90 04  c9
27b0 : 29 7f 09 40 a2 00 a0 03  96
27b8 : 85 6c 86 6d 06 6c 26 6d  fa
27c0 : 88 d0 f9 18 a5 6c 69 00  95
27c8 : 85 6c a5 6d a6 f1 30 03  5b
27d0 : 69 d0 2c 69 d8 65 f3 85  6d
27d8 : 6d 20 47 28 8e 03 ff a0  6e
27e0 : 07 b1 6c 48 88 10 fa 20  19
27e8 : 45 a8 a2 08 86 8d b6 b6  09
27f0 : 68 2a 48 90 06 20 58 28  a5
27f8 : 20 ab 23 ee 60 11 d0 03  6c
2800 : ee 61 11 c6 b6 d0 e9 18  86
2808 : a9 01 6d 62 11 8d 62 11  03
2810 : 90 03 ee 63 11 68 38 ad  da
2818 : 60 11 e9 08 8d 60 11 b0  fe
2820 : 03 ce 61 11 a2 08 c6 bd  06
2828 : d0 c4 38 ad 62 11 e9 04  7c
2830 : 8d 62 11 b0 03 ce 63 11  9f
2838 : 18 ad 60 11 69 0c 8d 60  4f
2840 : 11 90 03 ee 61 11 60 38  c9
2848 : ad 60 11 e9 04 8d 60 11  f7
2850 : b0 03 ce 61 11 4c 77 27  01
2858 : a0 2b 20 74 9a a2 03 bd  6b
2860 : 60 11 9d 54 11 9d 58 11  bc
2868 : ca 10 f4 a9 90 20 f3 9a  bc
2870 : 0e 54 11 2e 55 11 0e 56  75
2878 : 11 2e 57 11 a2 29 a0 ff  8e
2880 : 20 bb 28 a2 25 a0 ef 20  34
2888 : bb 28 a2 27 a0 ff 20 bb  e7
2890 : 28 a2 23 a0 ff 20 bb 28  26
2898 : a0 25 20 6d 9d a2 1f 20  2c
28a0 : 70 9d 8d 31 11 8c 32 11  c9
28a8 : a2 27 a0 29 20 7c 9d a2  cd
28b0 : 21 20 70 9d 8d 33 11 8c  81
28b8 : 34 11 60 a9 ff 20 ae 9d  b9
28c0 : 9d 31 11 98 9d 32 11 60  be
28c8 : 20 45 a8 a2 19 4c da cd  04
28d0 : 20 f4 87 e0 02 f0 26 e0  6a
28d8 : 03 f0 20 e0 04 f0 5b e0  6e
```

```
28e0 : 05 f0 5a 4c 63 69 02 85  12
28e8 : 4b 4b 3d f3 88 6f 80 00  ad
28f0 : 00 86 ef 00 00 00 88 15  7c
28f8 : 60 00 00 a2 ff 8e e6 28  0d
2900 : 20 86 03 d0 01 60 20 5c  8b
2908 : 79 20 d7 77 a2 e7 a0 28  b2
2910 : 20 03 8c 20 5c 79 20 d7  9b
2918 : 77 a2 ec a0 28 20 03 8c  d8
2920 : 20 5c 79 20 d7 77 a2 f1  78
2928 : a0 28 20 03 8c 20 5c 79  73
2930 : 20 d7 77 a2 f6 a0 28 4c  1c
2938 : 03 8c a9 38 2c a9 18 8d  7e
2940 : 52 26 60 20 86 03 c9 a4  b2
2948 : f0 17 20 96 26 20 86 03  22
2950 : d0 03 4c b2 26 20 86 03  8f
2958 : c9 2c f0 05 c9 a4 f0 01  9c
2960 : 60 48 20 80 03 a2 04 20  92
2968 : 7c 29 68 10 06 20 e0 22  be
2970 : 4c 55 29 20 f2 9d 20 b2  b7
2978 : 26 4c 55 29 ad e6 28 30  52
2980 : 03 4c 70 9e 86 9b 20 aa  b4
2988 : 29 20 97 29 20 86 03 f0  f0
2990 : 18 20 5c 79 20 cc 29 90  2d
2998 : 0c a6 9b a5 67 9d 31 11  dd
29a0 : a5 66 9d 32 11 e6 9b e6  aa
29a8 : 9b 60 20 86 03 d0 05 ba  8c
29b0 : e8 e8 9a 60 c9 2c f0 39  f3
29b8 : 20 d7 77 a9 e7 a0 28 20  3b
29c0 : 08 8a 20 4b af a9 ec a0  bc
29c8 : 28 4c e6 29 20 86 03 f0  19
29d0 : de c9 2c f0 1c 20 d7 77  cd
29d8 : a9 f1 a0 28 20 08 8a 20  54
29e0 : 4b af a9 f6 a0 28 20 12  3c
29e8 : 8a 20 4b af 20 b4 84 38  75
29f0 : 60 20 80 03 18 60 38 a9  9a
29f8 : 18 08 20 f4 87 e0 01 f0  20
2a00 : 0a e0 02 f0 03 4c 28 7d  47
2a08 : a9 80 2c a9 00 28 b0 02  3a
2a10 : 09 01 48 20 86 03 d0 0a  88
2a18 : 68 a8 a2 4e 8e 00 ff 4c  c8
2a20 : 00 44 20 5c 79 20 7b 87  6c
2a28 : a6 24 a4 25 20 bd ff a2  e3
2a30 : 08 a0 00 20 ba ff a9 01  e1
2a38 : a2 01 20 68 ff 20 45 a8  d7
2a40 : a9 04 8d 06 d5 68 10 03  f6
2a48 : a0 45 2c a0 04 84 [illegible] a2  d2
2a50 : 00 86 9e 29 01 d0 08 a9  6a
2a58 : 00 20 d5 ff 4c 69 2a 98  c8
2a60 : 18 69 40 a8 a9 9e 20 d8  14
2a68 : ff a2 07 8e 06 d5 90 03  a3
2a70 : aa d0 06 24 90 f0 05 a2  72
2a78 : 24 4c 3c 4d 60 20 86 03  a2
2a80 : c9 2c f0 0a a2 00 20 7c  80
2a88 : 29 20 86 03 f0 16 20 80  05
2a90 : 03 20 d0 26 e8 a9 03 85  8f
2a98 : 0d a5 05 48 86 05 20 ca  69
2aa0 : 58 68 85 05 60 8a 38 e9  3d
2aa8 : 0a 8d e3 2c 90 0c f0 16  51
2ab0 : aa ca f0 09 ca f0 0f ca  23
2ab8 : f0 03 4c 28 7d 20 1c 9e  c8
2ac0 : ca e8 d0 f6 f0 07 20 1c  11
2ac8 : 9e e0 45 b0 ed 8e e8 2c  8d
2ad0 : 20 86 03 d0 e5 ae e3 2c  ca
2ad8 : bd d6 2c 8d 63 2c bd da  01
2ae0 : 2c 8d df 2c 29 03 aa a0  e7
2ae8 : e0 a9 01 8c 66 2c 8d 6b  44
2af0 : 2c ca f0 13 18 98 6d 66  e9
2af8 : 2c 8d 66 2c ad 6b 2c 69  c4
2b00 : 01 8d 6b 2c ca d0 ed a0  54
2b08 : 78 a9 00 8c 77 2b 8d 7f  ec
2b10 : 2b a2 78 8e e2 2c 2c df  7c
2b18 : 2c 08 10 02 70 09 4e e2  db
2b20 : 2c 0e 77 2b 2e 7f 2b 28  72
2b28 : 30 09 4e e2 2c 0e 77 2b  34
2b30 : 2e 7f 2b ac 77 2b ad 7f  05
2b38 : 2b 8c 8e 2b 8d 90 2b 20  fd
2b40 : 45 a8 a9 00 20 bd ff a9  87
2b48 : 04 a2 04 a0 01 20 ba ff  ae
2b50 : 20 c0 ff 90 03 4c ba 2c  b8
2b58 : a2 04 20 c9 ff 20 17 fa  91
2b60 : 1b 33 18 0f 00 20 4a 2c  7f
2b68 : 20 00 8d e0 2c 8d e1 2c  a0
2b70 : f0 11 18 ad e0 2c 69 00  ba
2b78 : 8d e0 2c ad e1 2c 69 00  5b
2b80 : 8d e1 2c a9 ea a2 2c 8d  ce
2b88 : ad 2b 8e ae 2b a9 00 a0  85
2b90 : 00 8d e6 2c 8c e7 2c 20  8f
2b98 : 45 a8 a2 12 ad e1 2c 20  f7
2ba0 : cc cd e8 ad e0 2c 20 cc  cc
2ba8 : cd 20 d8 cd 8d ff ff ee  2c
2bb0 : ad 2b d0 03 ee ae 2b 38  09
2bb8 : ad e6 2c e9 01 8d e6 2c  91
2bc0 : ad e7 2c e9 00 8d e7 2c  0d
2bc8 : b0 df a2 00 a0 08 a9 00  01
2bd0 : 8d de 2c 3e ea 2c 2e de  26
2bd8 : 2c 3e 26 2d 2e de 2c 2c  35
2be0 : df 2c 10 02 70 29 3e 62  28
```

```
2be8 : 2d 2e de 2c 3e 9e 2d 2e  53
2bf0 : de 2c 2c df 2c 30 18 3e  0c
2bf8 : da 2d 2e de 2c 3e 16 2e  3a
2c00 : 2e de 2c 3e 52 2e 2e de  7d
2c08 : 2c 3e 8e 2e 2e de 2c ad  a3
2c10 : de 2c 20 6f 2c 88 d0 b6  b2
2c18 : e8 e0 3c d0 af ce e2 2c  ef
2c20 : f0 0b 20 6e f6 f0 06 20  bb
2c28 : 4a 2c 4c 72 2b 20 b0 2c  b9
2c30 : 20 b0 2c a9 1b 20 b2 2c  be
2c38 : a9 20 20 b2 2c a9 12 20  e8
2c40 : b2 2c 20 cc ff a9 04 4c  a0
2c48 : c3 ff 20 b0 2c ad e8 2c  55
2c50 : 8d e9 2c f0 0a a9 20 20  aa
2c58 : b2 2c ce e9 2c d0 f6 20  76
2c60 : 17 fa 1b 4c 00 a9 ff 20  d2
2c68 : b2 2c a9 ff 4c b2 2c 8e  c3
2c70 : e3 2c 8c e4 2c 2c df 2c  25
2c78 : 10 0b aa 70 05 bd c2 2c  68
2c80 : 50 03 bd d2 2c 8d de 2c  1e
2c88 : ad df 2c 48 29 03 8d e6  e8
2c90 : 2c 68 29 20 f0 08 ad de  02
2c98 : 2c 49 ff 8d de 2c ad de  de
2ca0 : 2c 20 b2 2c ce e6 2c d0  85
2ca8 : f5 ae e3 2c ac e4 2c 60  d6
2cb0 : a9 0d 20 d2 ff 24 90 d0  47
2cb8 : 01 60 20 42 2c a2 05 4c  be
2cc0 : 3c 4d 00 03 0c 0f 30 33  64
2cc8 : 3c 3f c0 c3 cc cf f0 f3  43
2cd0 : fc ff 00 0f f0 ff 4c 4c  87
2cd8 : 4c 4c 01 02 81 82 00 00  f7
2ce0 : 00 00 00 00 00 00 00 00  e1
2ce8 : 00 00 a9 04 8d 06 d5 a9  87
2cf0 : 00 85 a3 a2 12 20 cc cd  e1
2cf8 : e8 20 cc cd a2 1f 8e 00  3a
2d00 : d6 a9 40 85 a6 98 10 03  e1
2d08 : a2 45 2c a2 04 86 a4 98  e4
2d10 : a0 00 29 01 d0 15 b1 a3  df
2d18 : 2c 00 d6 10 fb 8d 01 d6  da
2d20 : c8 d0 f3 e6 a4 c6 a6 d0  e7
2d28 : ed f0 0e 20 dd cd 91 a3  ef
2d30 : c8 d0 f8 e6 a4 c6 a6 d0  38
2d38 : f2 a9 07 8d 06 d5 60 22  47
```

Listing 1. (Schluß)

```
Name : color-pack.key.m  1000 10ff
-----------------------------------
1000 : 02 0c 06 11 67 26 06 06  78
1008 : 05 0b 1b 40 09 09 09 1b  95
1010 : 40 1b 4a 44 cc 20 20 20  88
1018 : 1b 49 44 49 d2 0d 09 09  de
1020 : 09 1b 40 1b 4a 44 d3 20  81
1028 : 3a 22 22 9d 40 9d 9d 93  3e
1030 : 49 46 50 45 c5 28 32 29  12
1038 : 3d 31 54 c8 3a 47 d3 32  ce
1040 : 3a 50 4f cb 32 2c 30 3a  a9
1048 : 45 4c 53 45 49 46 50 45  c3
1050 : c5 28 32 29 3e 31 54 c8  2b
1058 : 50 4f cb 32 2c 31 3a 47  4d
1060 : d2 31 3a 47 d3 31 3a 3f  70
1068 : 43 c8 28 32 37 29 3b 43  90
1070 : c8 28 38 38 29 3b 3a 45  41
1078 : 4c 53 45 50 4f cb 32 2c  3e
1080 : 32 3a 47 d2 3a 3f 43 c8  38
1088 : 28 32 37 29 3b 43 c8 28  fe
1090 : 38 38 29 3b 3a 0d 43 4c  48
1098 : cf 34 3a 4f d0 34 2c 34  c2
10a0 : 2c 32 3a 43 cd 34 3a 4c  dc
10a8 : 49 53 54 09 09 09 3a 50  34
10b0 : d2 34 3a 43 4c cf 34 1b  de
10b8 : 4a 09 09 09 93 4c 49 53  52
10c0 : 54 0d 1b 49 4d 4f ce 0d  2f
10c8 : 93 52 55 4e 0d 41 d5 31  38
10d0 : 30 0d 91 91 91 1b 40 91  33
10d8 : 1b 40 91 4c 49 53 54 0d  9c
10e0 : 1b 49 4d 4f ce 0d 93 52  25
10e8 : 55 4e 0d 41 d5 31 30 0d  92
10f0 : 91 91 91 1b 40 91 00 00  a2
10f8 : ff ff 00 00 ff ff 00 63  be
```

Listing 2. In »color-pack.key.m« ist die neue Belegung der Funktionstasten gespeichert. Die Eingabe erfolgt mit dem MSE (Seite 158) im C64-Modus.

```
10 SCNCLR                                                    <FTJ>
20 PRINT "{YELLOW,CTRL+N}←←←←←←←←←←←←←←← COLOR-PA
   CK 1 ←←←←←←←←←←←←←←←←{2DOWN}"                            <L86>
30 :                                                         <IQB>
40 PRINT CHR$(159);"RAINER FRANZEN                           <O6C>
50 PRINT "EDGAR- ROV- STR. 19                                <AFQ>
60 PRINT "2000 HAMBURG 20                                    <TU4>
70 PRINT "TEL: 040/462008{GREEN,DOWN}                        <4BC>
80 :                                                         <NAT>
100 IF RGR(0)=5 THEN 160                                     <PHH>
110 PRINT " DIE 80-ZEICHEN-TASTE DRÜCKEN{CTRL+J}"            <2RE>
120 PRINT CHR$(27)"X";                                       <RQT>
130 PRINT "{CTRL+N} DIE 80-ZEICHEN-TASTE DRÜCKEN{
    CTRL+J}"                                                 <R86>
140 SYS DEC("FF3D")                                          <9VN>
150 :                                                        <0S3>
160 IF PEEK(211) AND 16 THEN 200                             <G2K>
170 PRINT "{CLR} DIE ASCII/DIN- TASTE DRUECKEN !"            <KC6>
180 SYS DEC("FF3D")                                          <97K>
190 :                                                        <6SR>
200 GRAPHIC 3,1                                              <52N>
210 GRAPHIC 0                                                <A7P>
220 PRINT "{CTRL+N} 80-ZEICHEN-MONITOR BENUTZEN !
    "                                                        <P6H>
230 GRAPHIC 5                                                <A3R>
240 :                                                        <3CD>
250 BLOAD "COLOR-PACK 1.M",ON B0                             <425>
260 BLOAD "COLOR-PACK.KEY.M",ON B0                           <9F7>
270 SYS DEC("23D2")                                          <EPE>
280 CLR                                                      <GE3>
290 FAST                                                     <DLJ>
```

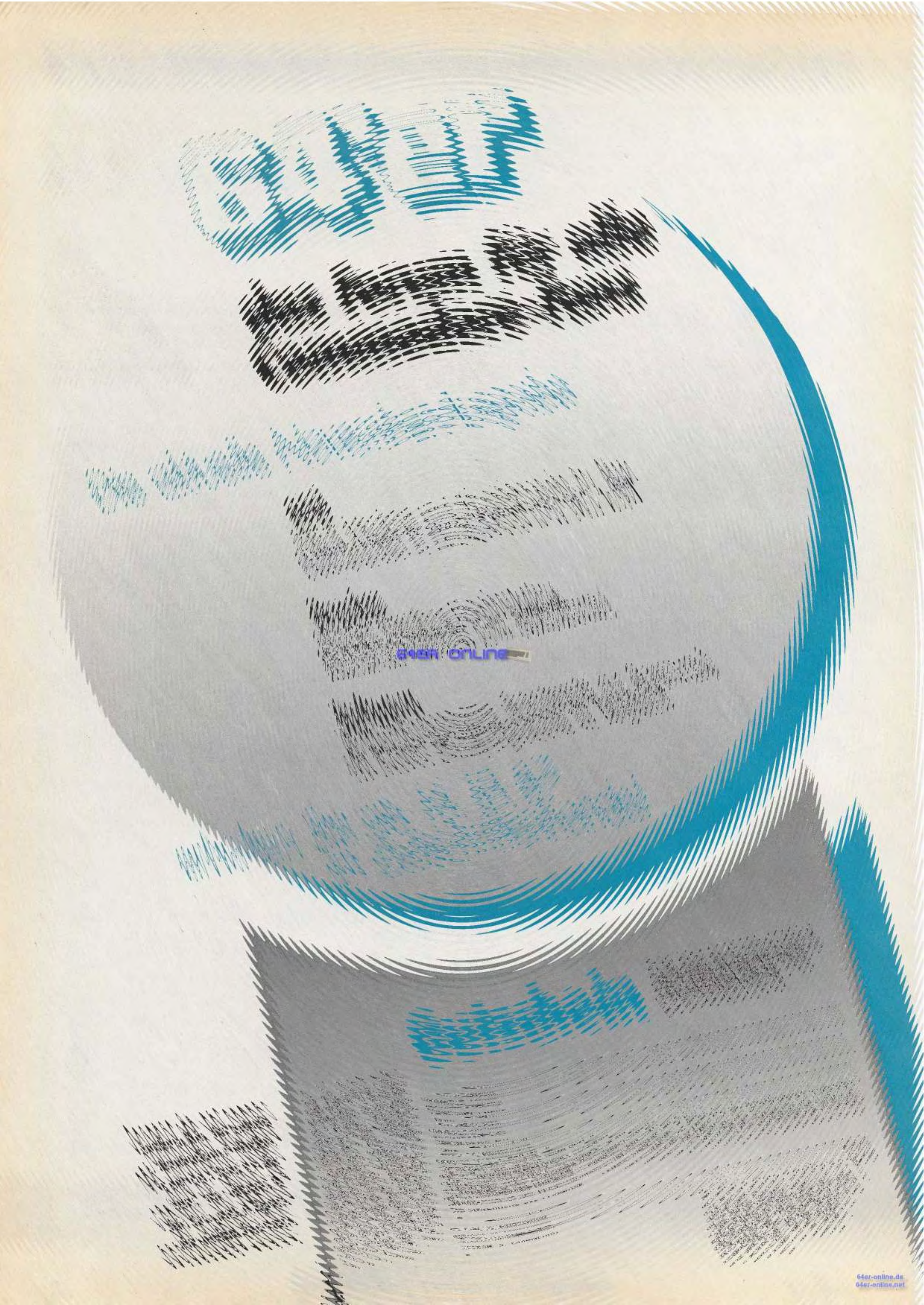
Listing 3. Der Lader von Color-Pack 1. Er hat natürlich den Namen »color-pack 1«. Der Checksummer 128 (Seite 158) ist als Eingabehilfe zu verwenden.

```
100 PRINT "←←←←←←←←←←←←←←←←←←←← CHAR-{5SPACE}BEFE
    HL ←←←←←←←←←←←←←←←←←←←←←"                               <F3I>
110 :                                                        <2SB>
120 TRAP 470                                                 <1TF>
130 GRAPHIC 1,8,7                                            <DFA>
140 COLOR 1,8,7                                              <V5U>
150 COLOR 2, 5,7                                             <ULS>
160 COLOR 3,11,7                                             <LQ3>
170 :                                                        <1S7>
180 CHAR 1, 90, 4 ,  0,"{RVSON}SCHRIFTDEMONSTRATI
    ON{RVOFF}",255,255                                       <VUP>
190 CHAR 1, 91, 4 ,  0,"{RVSON}SCHRIFTDEMONSTRATI
    ON{RVOFF}"                                               <0D9>
200 :                                                        <1C5>
210 CHAR 2, 40,20 ,  0," DAS IST DIE VOREINSTELLU
    NG "                                                     <QBS>
220 CHAR 2, 41,20 ,  0," DAS IST DIE VOREINSTELLU
    NG "                                                     <K8K>
230 :                                                        <0S3>
240 CHAR 3,110,36 ,  0,"DAS IST DIE NORMALE ZEICH
    ENGRÖßE{2SPACE}",127,255                                 <4DK>
250 :                                                        <3SF>
260 A$="DAS IST DIE SCHMALSCHRIFT BREITE,HÖHE: 92
    ,255"                                                    <VDT>
270 CHAR 1,110,52 ,  0, A$, 92,255                           <GS5>
280 :                                                        <5CL>
290 A$="DAS IST DIE NIEDRIG- UND SCHMALSCHRIFT BR
    EITE,HÖHE 95,224"                                        <4OB>
300 CHAR 1,110,72 ,  0, A$, 95,224                           <M9Q>
310 :                                                        <0S3>
320 A$="DAS IST DIE NIEDRIG- UND SCHMALSCHRIFT BR
    EITE,HÖHE 95,192"                                        <J8L>
330 CHAR 1,110,90 ,  0, A$, 95,192                           <95K>
340 :                                                        <2C9>
350 A$="DAS IST DIE NIEDRIG- UND SCHMALSCHRIFT BR
    EITE,HÖHE 95,176"                                        <54C>
360 CHAR 1,110,108 ,  0, A$, 95,176                          <9TQ>
370 :                                                        <3SF>
380 CHAR 1, 20,200, 90,"VOREINSTELLUNG: ",,,255,2
    39                                                       <DV4>
390 :                                                        <4SJ>
400 CHAR 1, 36,200, 90,"Y BREITE HÖHE{2SPACE}255,
     239"                                                    <LU9>
410 :                                                        <7SV>
420 CHAR 1, 52,200, 90,"Y BREITE HÖHE{2SPACE}200,
     220",,,200,220                                          <JUG>
430 :                                                        <6SR>
431 CHAR 1,420,140,180,"DIESER TEXT STEHT KOPF",2
    55,255                                                   <BQU>
432 CHAR 1,420,160,180,"EIN SCHMALER TEXT STEHT K
    OPF",150,255                                             <7CV>
433 CHAR 1,110,170,340,"SCHRÄGE TEXTE GEHEN AUCH"            <J0U>
440 :                                                        <5CL>
450 GET KEY A$                                               <LSD>
460 :                                                        <4CH>
470 GRAPHIC                                                  <NBG>
480 IF ER<1 THEN 500                                         <3SO>
490 PRINT ERR$(ER) " IN ZEILE "EL                            <7RK>
500 END                                                      <T7M>
```

Listing 4. Das Demoprogramm erzeugt Bild 2. Bitte mit dem Checksummer 128 (Seite 158) eingeben.

```
60 TRAP 10000                                                <JL0>
61 :                                                         <G20>
100 REM "←←←←←←←←←←← NEUE INTERNE ACHSENSCALIERUNG
     ←←←←←←←←←←←←←←←:                                        <2C9>
110 :                                                        <2SB>
120 X1=-3*π:  X2=3*π                                         <VA1>
130 :                                                        <3SF>
140 Y1=-1.5:  Y2=2.5                                         <B9J>
150 :                                                        <0S3>
160 REM "←← BERECHNUNG DER SCALIERUNGSFAKTOREN UN
    D -SUMMANDEN ←←:                                         <1C5>
170 :                                                        <1S7>
180 XF=(479-0)/(X2-X1)           : REM " STEIGUNG            <MG8>
190 XS=0-XF*X1                   : REM " ACHSENABS
    CHNITT                                                   <J96>
200 :                                                        <1C5>
210 YF=(0-239)/(Y2-Y1)                                       <195>
220 YS=239-YF*Y1                                             <V3G>
230 :                                                        <0S3>
240 SCALE 3, XF,XS  ,YF,YS       : REM " NEUES KOO
    RDINATENSYSTEM                                           <EDH>
250 :                                                        <3SF>
260 GRAPHIC 1,8,7                : REM " GELB/BLAU
     INS GESAMTE FARBRAM                                     <5I8>
270 :                                                        <2SB>
280 XA=-2.1*π: XB=+2.1*π         : REM " START- UN
    D ENDWERT                                                <VS3>
290 :                                                        <5SN>
300 DRAW ,XA-.5,0 TO XB+.5,0     : REM " KOORDINAT
    ENACHSEN ZEICHNEN                                        <JQD>
310 DRAW ,0,-1.2 TO 0,1.2                                    <SUT>
320 :                                                        <1C5>
330 DRAW ,XA,SIN(XA)                                         <FND>
340 FOR X=XA TO XB STEP 2/XF     : REM " STEP: JED
    ES 2. PIXEL IN X- RICHTUNG                               <B78>
350 : DRAW TO X,SIN(X)                                       <9BP>
360 NEXT X                                                   <9LC>
370 :                                                        <3SF>
380 COLOR 2,5,4                  : REM " VIOLETT/
    TÜRKIS                                                   <8V7>
390 CHAR 2,120,20,,"{RVSON} SINUSKURVE ",255,255             <P8S>
391 CHAR 2,121,20,,"{RVSON} SINUSKURVE "                     <CFU>
400 :                                                        <7CT>
410 COLOR 3,11,1                 : REM " ROT/SCHWA
    RZ                                                       <P8T>
420 BOX 3, 3,3,475,235                                       <RJQ>
430 :                                                        <6SR>
1000 SCALE 2               : REM "←←←←← LOCATE - AN
     WENDUNG ←←←←←←"                                         <1MD>
1010 FOR X=10 TO 469                                         <5HN>
1020 : LOCATE X,200,C                                        <5KU>
1030 : DRAW 1                                                <PI5>
1040 : IF C THEN BEGIN                                       <EDL>
1050 :      DRAW 1,X-4,202 TO X+4,198                        <H5B>
1060 :      DRAW 1,X-4,198 TO X+4,202                        <MP1>
1070 :    BEND : ELSE  BEGIN                                 <ROH>
1080 :      DRAW 0                                           <905>
1090 :    BEND                                               <3KU>
1100 NEXT                                                    <A90>
1110 GET KEY A$                                              <78D>
1120 :                                                       <105>
10000 GRAPHIC                                                <VJG>
10010 IF ER<1 THEN 10030                                     <4AD>
10020 PRINT ERR$(ER)" IN ZEILE "EL                           <JBU>
10030 LIST 100 - 280                                         <AAO>
```

Listing 5. »SCALE« definiert jeden beliebigen Bereich. Bitte mit dem Checksummer 128 (Seite 158) eingeben.



Apfelberge

Fraktale Grafik ist unter Computerkünstlern bestens bekannt. Das folgende Programm erlaubt eine besondere Art der Apfelmännchen im Multicolor-Modus des C 128 darzustellen.

Fraktale Grafiken, die man auch »Apfelmännchen« nennt, entstehen trotz ihrer komplexen Schönheit aus einer einfach scheinenden mathematischen Funktion. Lesen Sie dazu auch den neuen Fraktal-Kurs des 64'er-Magazins, der in Ausgabe 9/87 beginnt.

In die Zweidimensionalität gezwungen, wirken fraktale Bilder wie kunstvolle Gemälde, deren Farben durch die Berechnungstiefe eines jeden Punktes bestimmt sind. Eine ebenso interessante Version sind jedoch dreidimensionale Apfelmännchen. Hier wird die Iterationstiefe eines Punktes nicht farbmäßig, sondern durch die dritte Dimension dargestellt. Das Ergebnis sind eigenartige und mitunter zufällig scheinende Hügellandschaften, wie sie oftmals in Spielen wie »Koronis Rift« Verwendung finden.

Das Basic-Programm »apfelberg.mod« in Listing 1 ist eine für den C 128 aufbereitete Version des Programmes »apfelberge«, das bereits in Ausgabe 1/87 des 64'er-Magazins für den C 64 veröffentlicht wurde. Verwenden Sie für die Eingabe bitte den Checksummer 128, den Sie in diesem Sonderheft auf Seite 158 finden.

Mit den komfortablen Befehlen des Basic 7.0 zaubert dieses Programm herrliche Landschaften auf den 40-Zeichen-Bildschirm des C 128, wie etwa die Grafik in Bild 1. Verschiedene Parameter erlauben dabei die Wahl der verwendeten Farben sowie die Veränderung des Bildausschnittes und der Iterationstiefe.

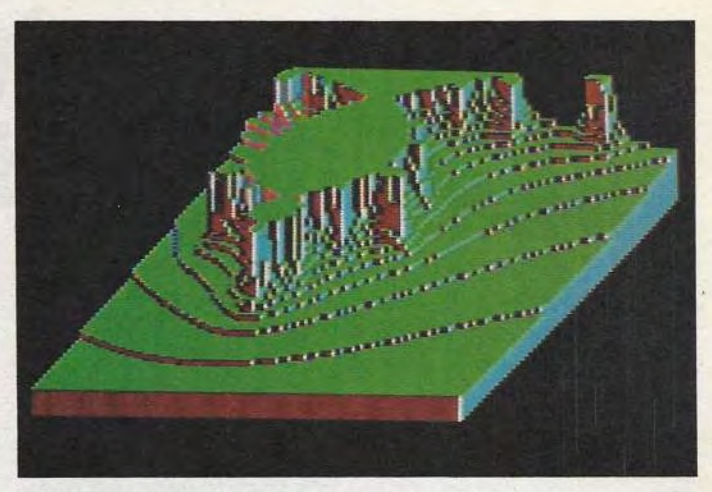


Bild 1. Die »Grundlandschaft« der Apfelberge

Trotz des Befehles FAST ist die Berechnung eines solchen Bildes jedoch sehr zeitaufwendig. Je nach gewählter Iterationstiefe ergeben sich Rechenzeiten bis in den Stundenbereich. Aus diesem Grund sollte das fertige Bild später auf Diskette gespeichert werden. Das Grafikbild ab Adresse 8192 bis 16383 und der dazugehörige Farbspeicher von Adresse 7168 bis 8191 werden dabei getrennt in zwei Dateien abgelegt, um eine eventuelle Bearbeitung des Bildes mit einem Zeichenprogramm im C 64-Modus zu erleichtern. Die Datei mit der Farbinformation ist durch das Kürzel »-F« gekennzeichnet. Soll die Grafik wieder im C 128-Modus geladen werden, sind demnach folgende Anweisungen notwendig:

```
BLOAD"⟨dateiname⟩", ON B0
BLOAD"⟨dateiname⟩-F", ON B0
```

Der Befehl »GRAPHIC 3« macht das Bild anschließend wieder sichtbar. (Rene Gagnaux/Michael Thomas)

```
100 REM APFELBERGE, NACH FLORIAN MUELLER,1986, 64
    'ER, HEFT 1/87, S.80                            <IIA>
110 REM URSPRUENGLICH FUER C16/116/PLUS 4, ABGEAE
    NDERT FUER C 128                                <IQB>
120 REM ABAENDERUNG 14.12.1986, R.GAGNAUX           <I28>
130 REM TEXTAUSGABE AUF 80-ZEICHENBILDSCHIRM        <IA9>
140 REM GRAFIKAUSGABE AUF 40-ZEICHENBILDSCHIRM, G
    RAPHIC-MODUS 3                                  <JIE>
150 :                                               <0S3>
160 :                                               <1C5>
200 SCNCLR : GRAPHIC 0,1: GRAPHIC 1,1: GRAPHIC 3,
    1: GRAPHIC 5,1                                  <GHR>
400 REM VOREINSTELLEN DER VARIABLEN                 <G20>
401 XC=1: YC=0: T=20: S=60: XL=-.15: XR=.26: YO=.
    47: YU=.9: XM=105: YM=105                       <3KP>
402 F1=6: REM FARBE DER HORIZONTALEN FLAECHEN       <3S5>
403 F2=4: REM FARBE DER VERTIKALEN SEITENFLAECHEN   <4CR>
404 F3=11: REM FARBE DER VERTIKALEN VORDERFLAECHE
    N                                               <LU2>
405 HG=7: REM HINTERGRUNDFARBE GRAPHIKBILDSCHIRM    <LQS>
410 PRINT TAB(10)"HINTERGRUND.....";HG              <PBV>
420 PRINT TAB(10)"FARBE 1.........";F1              <P7O>
430 PRINT TAB(10)"FARBE 2.........";F2              <LRR>
440 PRINT TAB(10)"FARBE 3.........";F3              <IFL>
450 PRINT TAB(10)"X KOMPLEX.......";XC              <CFT>
460 PRINT TAB(10)"Y KOMPLEX.......";YC              <CNU>
470 PRINT TAB(10)"RECHENTIEFE.....";T               <JQU>
480 PRINT TAB(10)"MAXIMALE HOEHE..";S               <L2K>
490 PRINT TAB(10)"X LINKS.........";XL              <UN6>
500 PRINT TAB(10)"X RECHTS........";XR              <GR7>
510 PRINT TAB(10)"Y OBEN..........";YO              <MBL>
520 PRINT TAB(10)"Y UNTEN.........";YU              <RB1>
530 PRINT TAB(10)"AUSMASS X.......";XM              <474>
540 PRINT TAB(10)"AUSMASS Y.......";YM              <6VB>
550 PRINT : PRINT "DIE VOREINGESTELLTEN WERTE KOE
    NNEN MIT <RETURN> - OHNE PARAMETEREINGABE -"    <OVT>
560 PRINT "UEBERNOMMEN WERDEN !!"                   <LS5>
570 PRINT                                           <5CU>
580 :                                               <2C9>
590 REM  BEGINN EINGABE                             <IQB>
600 INPUT "HINTERGRUND";HG                          <16S>
610 INPUT "FARBE 1{4SPACE}";F1                      <LL8>
620 INPUT "FARBE 2{4SPACE}";F2                      <DM9>
630 INPUT "FARBE 3{4SPACE}";F3                      <5N9>
640 INPUT "X KOMPLEX{2SPACE}";XC                    <S7A>
650 INPUT "Y KOMPLEX{2SPACE}";YC                    <SEB>
660 INPUT "MAX. TIEFE ";T                           <FK9>
670 INPUT "MAX. HOEHE ";S                           <GJQ>
680 INPUT "X LINKS{4SPACE}";XL                      <4HD>
690 INPUT "X RECHTS{3SPACE}";XR                     <EU2>
700 INPUT "Y OBEN{5SPACE}";YO                       <RBD>
710 INPUT "Y UNTEN{4SPACE}";YU                      <FM0>
720 DO : INPUT "AUSMASS X";XM: LOOP WHILE XM=0      <ES1>
730 DO : INPUT "AUSMASS Y";YM: LOOP WHILE YM=0      <RCN>
740 PRINT "ALLES RICHTIG (J/N) ?"                   <9IS>
750 DO : GET KEY A$: LOOP UNTIL A$="J" OR A$="N"    <4GQ>
760 :                                               <7CT>
800 REM BEGEIN GRAFIK                               <M2O>
801 IF A$="N" THEN RUN : ELSE DX=(XR-XL)/XM: DY=(
    YU-YO)/YM                                       <4BO>
810 GRAPHIC 1,1: GRAPHIC 5,1                        <VR4>
815 PRINT "DIE AUSGABE DER GRAFIK ERFOLGT AUF DEM
     40-ZEICHEN-BILDSCHIRM !!"                      <EKM>
820 GRAPHIC 3,1: COLOR 1,F1: COLOR 4,HG: COLOR 2,
    F2: COLOR 3,F3: COLOR 0,HG                      <11N>
830 FOR N=0 TO YM: Y1=YO+N*DY: FOR M=0 TO XM: X=X
    L+M*DX: Y=Y1: K=0                               <OBV>
840 DO : X2=X*X: Y2=Y*Y: Y=2*X*Y-YC: X=X2-Y2-XC:
    K=K+1: LOOP WHILE((K<T) AND(X2+Y2<S))           <MFV>
850 U=M+53-N/2: U1=U+1: V=N+80: V1=V-3*(K-1)        <TE0>
855 IF V1<0 THEN V1=0                               <QHF>
860 DRAW 3,U,V TO U,V1: DRAW 2,U1,V TO U1,V1: DRA
    W 1,U,V1 TO U1,V1                               <DMG>
870 NEXT M                                          <JM6>
875 PRINT "{HOME,5DOWN,WHITE}DIE GRAFIK IST ZU ";
    (INT((100*(N/(YM+1)))*100))/100;" %{SHF.SPACE
    }GEZEICHNET{10SPACE}"                           <4TH>
880 NEXT N                                          <BL7>
900 REM BILD SPEICHERN                              <MIQ>
910 GRAPHIC 5,1                                     <H2R>
920 PRINT : PRINT "BILD ABSPEICHERN ?{SHF.SPACE}(
    J/N)"                                           <MEG>
930 PRINT "(DIRECTORY MIT TASTE 'D')"               <BQ1>
940 GET KEY A$: IF A$="D" THEN DIRECTORY : GOTO 9
    20                                              <3NE>
950 IF A$="J" THEN BEGIN                            <9D4>
960 INPUT "BILDNAME, MAX. 13 ZEICHEN";B$            <7BE>
965 IF LEN(B$)>13 THEN GOTO 960                     <H1V>
970 BSAVE(B$),ON B0,P8192 TO P16383                 <JLH>
975 BSAVE(B$)+"-F",ON B0,P7168 TO P8191             <I9M>
980 BEND                                            <IME>
990 END                                             <V3U>
```

Listing 1. Das modifizerte Programm »apfelberg.mod«, das die fraktalen Gebirge erzeugt. Bitte mit Checksummer 128 (Seite 158) eingeben.

»Echte« Windows im 80-Zeichen-Modus

Damit Sie Ihre Programme professioneller und übersichtlicher gestalten können, stellen wir hier eine Erweiterung zum Abtippen vor, die zehn »echte« Windows bearbeiten kann.

Das Programm »Window-Tech 80« ist eine Betriebssystem-Erweiterung für den 80-Zeichen-Modus des C128, die nicht nur 10 (!) Windows gleichzeitig bearbeitet, sondern es können auch noch die Bildschirminhalte gerettet werden. »Ganz nebenbei« wird der 80-Zeichen-Bildschirm um zwei weitere Bildschirmzeilen erweitert, von denen die zweite am unteren Bildschirmrand die aktuelle Uhrzeit (inklusive Alarm) und den aktuellen Programmzählerinhalt der CPU 8502 laufend anzeigt (Bild 1); die erste Zusatz-Zeile bleibt leer und dient nur zur Verschönerung des Designs, damit die Statuszeile nicht an der untersten regulären Bildschirmzeile »klebt«.

10 Windows gleichzeitig

Das Programm, das die genannten Eigenschaften hat, finden Sie in Listing 1; es wird mit dem MSE im C64-Modus eingegeben, aber im C128-Modus mit BLOAD"WINDOW-TECH 80",ON B0 geladen und mit SYS DEC("1BD6") gestartet. Danach ist diese Erweiterung aktiviert, was sich an der Einblendung der Statuszeile bemerkbar macht.

Statuszeile als 27. Bildschirmzeile

Die Anzeige des PC (Programmzähler des Prozessors) ist für Maschinenprogrammierer beim Debugging (Fehlersuche) nützlich; bleibt der PC stehen, so durchläuft das Maschinenprogramm entweder eine Schleife, die immer wieder ihre eigene Adresse anspringt ($5000 jmp $5000), oder es liegt ein Prozessorabsturz vor. Beim Hängenbleiben eines Programms in einer größeren Endlosschleife kann man die kritische Adresse zumindest erahnen.

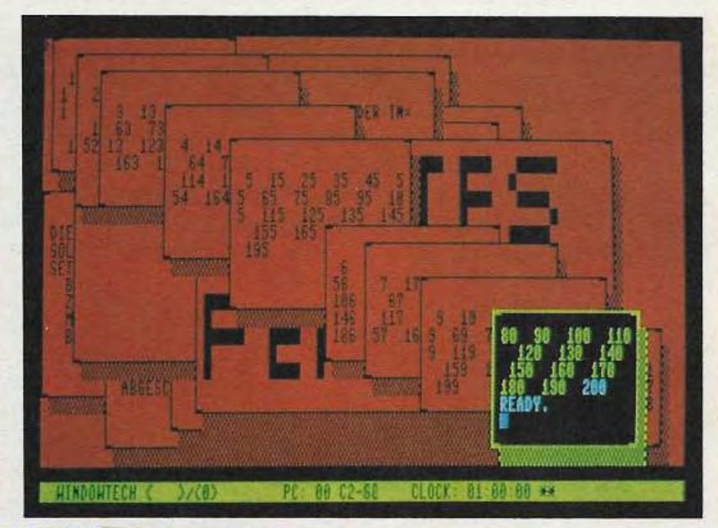


Bild 1. [illegible] Demo für die Windows. Das rechte untere Window ist momentan aktiv.

Rechts neben dem PC befindet sich eine Echtzeituhr und noch einmal rechts davon die Alarmzeit, die durch Einstellen aktiviert wird.

Nach dem Start des Programms befindet man sich im Fenster mit der Nummer »#1«. Alle zehn Fenster sind in ihrer Größe frei wählbar, sind aber nach dem Start schon voreingestellt. Neu einstellen lassen Sie sich auf die herkömmliche Art und Weise mittels ESC-Funktionen, oder

```
NAME : WINDOW-TECH 80      1300 1BF7
-------------------------------------
1300 : B0 B1 B2 B3 B4 B5 B6 B7  EF
1308 : B8 B9 81 82 83 84 85 86  CD
1310 : 48 29 0F AA BD 00 13 A8  7F
1318 : 68 4A 4A 4A 4A AA BD 00  72
1320 : 13 60 BA 86 FB E8 E8 E8  5F
1328 : E8 E8 9A 68 85 FC AA 68  F4
1330 : 85 FD 48 8A 48 A6 FB 9A  F6
1338 : A5 FD C9 CD F0 10 C9 6A  93
1340 : F0 0C C9 81 F0 08 C9 C5  DB
1348 : F0 04 C9 E1 D0 03 4C 98  70
1350 : 02 A2 12 20 DA CD 85 06  6A
1358 : E8 20 DA CD 85 07 A9 42  7C
1360 : 20 CC CD CA A9 08 20 CC  A8
1368 : CD 68 48 20 10 13 20 CA  2F
1370 : CD 98 20 CC CD A9 A0 20  18
1378 : CC CD A5 FD 20 10 13 20  63
1380 : CA CD 98 20 CC CD A9 AD  98
1388 : 20 CC CD A5 FC 20 10 13  6E
1390 : 20 CA CD 98 20 CC CD EE  19
1398 : F6 13 D0 4A A2 13 A9 55  A9
13A0 : 20 47 14 AD 0B DC 0A 08  EE
13A8 : 4A 28 90 04 F8 69 11 D8  7C
13B0 : 20 10 13 20 07 16 98 20  65
13B8 : 0D 16 A9 BA 20 CC CD AD  8D
13C0 : 0A DC 20 10 13 20 13 16  ED
13C8 : 98 20 19 16 A9 BA 20 CC  04
13D0 : CD AD 09 DC AC 08 DC 20  10
13D8 : 10 13 20 CA CD 98 20 CC  8F
13E0 : CD A9 D0 20 31 16 A2 12  2C
13E8 : A5 06 20 CC CD E8 A5 07  FB
13F0 : 20 CC CD 4C 98 02 F9 A0  36
13F8 : A0 97 89 8E 84 8F 97 94  E4
1400 : 85 83 88 A0 BC A0 A0 A0  12
1408 : BE AF BC A0 BE A0 A0 A0  96
1410 : A0 A0 A0 A0 A0 90 83 BA  4F
1418 : A0 20 20 A0 20 20 A0 20  AA
1420 : 20 A0 A0 A0 A0 83 8C 8F  44
1428 : 83 8B BA A0 20 20 A0 20  F9
1430 : 20 A0 20 20 A0 AA AA A0  F8
1438 : A0 A0 A0 A0 A0 A0 A0 A0  37
1440 : A0 A0 A0 A0 A0 A0 A0 20  3E
1448 : CC CD CA A9 08 4C CC CD  94
1450 : A9 1B A2 06 20 CC CD A2  D5
1458 : 12 A9 10 20 CC CD A9 00  29
1460 : E8 20 CC CD E8 A9 10 20  A1
1468 : CC CD 8D 2F 0A A2 00 8A  2F
1470 : 48 A0 00 A9 05 20 CA CD  56
1478 : C8 D0 FA 68 AA E8 E0 0A  FD
1480 : D0 ED A2 12 A9 07 20 CC  1F
1488 : CD A9 D0 E8 20 CC CD A0  5C
1490 : 00 A9 64 20 CA CD C8 C0  42
1498 : 50 D0 F8 A2 12 A9 08 20  B2
14A0 : CC CD A9 20 E8 20 CC CD  20
14A8 : A0 00 B9 F7 13 20 CA CD  AF
14B0 : C8 C0 50 D0 F5 A2 07 A9  EA
14B8 : 21 4C CC CD A9 02 8D 00  CD
14C0 : FF 4C 22 13 A9 00 8D 00  A1
14C8 : FF 4C 65 FA A0 00 B9 BC  10
14D0 : 14 99 90 02 C8 C0 10 D0  8A
14D8 : F5 78 A9 90 8D 14 03 A9  5F
14E0 : 02 8D 15 03 58 60 A0 00  5A
14E8 : B9 00 02 C9 53 F0 0B 4C  DD
14F0 : 8C 15 FE C9 41 F0 FE 6C  70
14F8 : 8C 16 EA AD 01 02 C9 45  D1
1500 : F0 03 6C 8C 16 AD 03 02  FD
1508 : C9 AA F0 7B C9 AB F0 69  62
1510 : AD 0F DC 29 7F 8D 0F DC  FB
1518 : AD 0E DC 09 80 8D 0E DC  8B
1520 : AD 03 02 AE 04 02 20 4C  0F
1528 : 15 8D 0B DC AD 05 02 AE  CB
1530 : 06 02 20 4C 15 8D 0A DC  69
1538 : AD 07 02 AE 08 02 20 4C  69
1540 : 15 8D 09 DC A9 00 8D 08  DB
1548 : DC 4C 74 15 A0 00 D9 EE  59
1550 : 15 F0 07 C8 C0 10 D0 F6  76
1558 : A0 00 98 0A 0A 0A 0A 85  84
1560 : 60 8A A0 00 D9 EE 15 F0  79
1568 : 07 C8 C0 10 D0 F6 A0 00  4D
1570 : 98 05 60 60 A2 FF 4C 3F  89
1578 : 4D 78 A9 FA 8D 15 03 A9  AC
1580 : 65 8D 14 03 58 D0 ED 20  15
1588 : CC 14 D0 E8 C9 57 D0 03  50
1590 : 4C 37 16 C9 41 F0 03 6C  B7
1598 : 8C 16 86 07 A2 12 20 95  18
15A0 : 16 E8 A9 61 20 CC CD A0  A2
```

mit eigens dafür eingerichteten ESC-Sequenzen. Beim Verlassen eines Fensters wird die Cursorposition zwischengespeichert; nach Einschalten dieses Fensters befindet sich der Cursor also wieder an der alten Position.

Zusätzliche Basic-Befehle für den Direktmodus

Um mit der Echtzeituhr möglichst bequem arbeiten zu können, wurden zwei zusätzliche Basic-Befehle - SET und ALARM - geschaffen, die nur im Direktmodus funktionieren.

SET dient dem Einstellen der Uhrzeit und wird von einer sechsstelligen Angabe (ähnlich der Basic-Variablen TI$) gefolgt. SET083410 setzt die Uhr beispielsweise auf 8 Uhr 34 und 10 Sekunden. Um eine Zeit zwischen 12 und 24 Uhr einzustellen, muß zur ersten Ziffer 8 addiert werden (Beispiel: SET914507 setzt die Uhr auf 23 Uhr 45 und 7 Sekunden, nicht jedoch SET234507!).

SET– schaltet die Anzeige der Statuszeile ab, die Uhr läuft aber intern weiter.

SET+ dient zur Rückgängigmachung von SET–.

ALARM stellt die Alarmzeit gemäß einer vierstelligen Angabe (ALARM0834 für Alarm um 8 Uhr 34, ALARM2345 für Alarm um 23 Uhr 45). Der Alarm dauert 1 Minute und wird mit <ESC H> abgestellt.

Zusätzliche ESC-Sequenzen

Außer den herkömmlichen ESC-Funktionen stellt das Programm »Window-Tech 80« auch noch einige weitere bereit, die zur Ausreizung seiner besonderen Möglichkeiten dienen.

<ESC ?> fragt nach einer Fensterauswahl; durch Drücken einer Zifferntaste (0 bis 9) wird dadurch das gewünschte Fenster aktiviert. Eine andere Taste verläßt den Auswahlmodus und behält das alte Fenster bei.

<ESC> und nachfolgendes Drücken einer Zifferntaste (0 bis 9) wählt ein Fenster direkt an.

Wird <ESC> vom Kleiner-als-Symbol (<) gefolgt, so definiert man damit die linke obere Ecke des Fensters, löscht dieses und stellt es umrandet dar.

Dasselbe bewirkt das Auslösen von <ESC> und dem Größer-als-Symbol (>) für die rechte untere Ecke.

Natürlich läßt sich die Fenstertechnik von »Window-Tech 80« auch in Basic verwenden. Auf folgende Weise wird ein beliebiges Fenster aufgerufen:

```
10 INPUT FENSTER:GOSUB 1000: REM FENSTER = 0 BIS 9
20 END
1000 PRINT CHR$(27);CHR$(48+FENSTER);:RETURN:
     REM FENSTER AUFRUFEN
```

Ebenso ist es für Maschinenprogrammierer möglich, die ESC-Sequenzen für die Fensterbehandlung mittels BSOUT oder PRIMM zu verwenden.

Eine weitere Möglichkeit besteht darin, die ESC-Sequenzen <ESC 0> bis <ESC 9> neu zu definieren, wobei <ESC ?> weiterhin funktioniert (!). Die Adreßtabelle für die ESC-Routinen befindet sich im Speicher ab $16c4 ($16c4 bis $16d3: Low-Bytes; $16d4 bis $16e3: High-Bytes).

Fensterparameter und -inhalte lassen sich speichern

In Basic erlaubt folgender Befehl das Speichern aller Fensterdaten einschließlich der Cursor-Koordinaten:

```
BSAVE"NAME",ON B0,P6011 TO P6071
```

Mit BLOAD"NAME",ON B0 werden diese wieder eingelesen.

Ein anderer Befehl speichert die Fensterinhalte:

```
BSAVE"NAME",ON B1,P44800 TO P65279
```

BLOAD"NAME",ON B1 lädt diese Datei wieder.

Die Lade- und Speicherbefehle sind für Maschinenprogrammierer leicht in die Kernel-Aufrufe von SETBNK, LOAD und SAVE zu übertragen.

Abschließend wünschen wir Ihnen viel Spaß mit dieser wirklich sinnvollen Erweiterung des C128. Im Literaturverzeichnis finden Interessierte noch einige Hinweise auf weiterführende Informationen zum Thema »Window-Programmierung auf dem C128«. (Heino Cunze/F. Müller)

Literaturhinweise:

Windows – Fenster zum neuen Bedienungskomfort, 64'er-Sonderheft 7/86 (PEEKs & POKEs), S. 110-117.
Dieser Artikel bezieht sich auch auf C16 und C64.

VDC – 80 Zeichen ist nicht alles, Artikelserie in den 64'er-Stammheften 3/87, 5/87, 7/87 und 8/87. Dort erfahren Sie auch alles über die Programmierung zusätzlicher Bildschirmzeilen.


```
15A8 : 00 B9 05 02 20 CD 15 20  0B
15B0 : CA CD C8 C0 02 F0 06 C0  EC
15B8 : 04 D0 EE F0 B7 A9 BA 20  F2
15C0 : CA CD 4C B7 15 B2 B2 B3  94
15C8 : B4 B1 B2 B1 B8 48 29 3F  29
15D0 : 4C FE 15 68 60 A9 15 A2  DB
15D8 : E6 D0 04 A9 4D A2 3F 8D  5F
15E0 : 01 03 8E 00 03 60 A9 00  E0
15E8 : 8D 00 FF 4C E6 14 30 31  31
15F0 : 32 33 34 35 36 37 38 39  E0
15F8 : 41 42 43 44 45 46 09 80  5F
1600 : 99 C5 15 68 09 80 60 8D  FF
1608 : C9 15 4C CA CD 8D CA 15  67
1610 : 4C CC CD 8D CB 15 4C CA  13
1618 : CD 8D CC 15 4C CC CD A0  25
1620 : 00 B9 C5 15 D9 C9 15 D0  F3
1628 : 0D C8 C0 04 D0 F3 4C 8E  45
1630 : C9 8D F6 13 D0 E9 60 AD  19
1638 : 01 02 C9 AA D0 03 4C 11  7A
1640 : 17 C9 AB D0 03 4C FF FF  D3
1648 : C9 31 F0 0F C9 32 F0 24  02
1650 : 4C 3F 4D 00 28 27 4F EA  5E
1658 : EA EA EA C6 F8 EA EA A2  22
1660 : 00 BD 53 16 85 E6 BD 55  08
1668 : 16 85 E7 20 86 16 8D 03  94
1670 : FF 4C BA 4D C6 F8 EA EA  A3
1678 : A2 01 BD 53 16 85 E6 BD  19
1680 : 55 16 85 E7 D0 E5 20 B4  65
1688 : B8 E6 F8 60 3F 4D EA EA  DD
1690 : 41 4C 41 52 4D A0 00 B9  DF
1698 : 00 02 D9 90 16 D0 0A C8  C3
16A0 : C0 05 D0 F3 A9 08 4C CC  3B
16A8 : CD A6 07 6C 8C 16 C9 40  39
16B0 : B0 0F E9 2F 30 0B AA EA  24
16B8 : BD D4 16 48 BD C4 16 48  59
16C0 : 60 4C C1 C9 E3 E3 E3 E3  A4
16C8 : E3 E3 E3 E3 E3 E3 A1 A1  39
16D0 : 14 A1 16 A1 16 16 16 16  05
16D8 : 16 16 16 16 16 16 19 19  EA
16E0 : 1B 19 1B 19 CA 10 02 A2  EC
16E8 : 09 4C EA 19 16 85 E6 BD  9A
16F0 : 55 16 85 E7 A5 EB F0 02  30
16F8 : C6 EB 20 7D FF 8D 00 60  99
1700 : A9 AE A2 16 8D 38 03 8E  2F
1708 : 39 03 60 A9 C1 A2 C9 D0  0A
1710 : F3 20 00 17 4C 74 15 20  F3
1718 : 0B 17 4C 74 15 A2 12 A9  52
1720 : 10 20 CC CD A9 00 E8 20  AC
1728 : CC CD A2 00 8A 48 A0 00  F1
1730 : A9 48 20 CA CD C8 D0 FA  BB
1738 : 68 AA E8 E0 08 D0 ED A2  4F
1740 : 12 A9 17 20 CC CD A9 D0  74
1748 : E8 20 CC CD A0 D0 A9 05  6E
1750 : 20 CA CD C8 D0 FA 60 20  08
1758 : 7D FF 46 41 4C 53 43 48  8C
1760 : 45 20 45 49 4E 47 41 42  D9
1768 : 45 0D 00 A2 FF 4C 3F 4D  82
1770 : A7 AF B7 BF C7 CF D7 DF  EF
1778 : E7 EF F7 04 0C 04 07 02  D6
1780 : 04 06 0A 08 02 04 0C 11  9E
1788 : 0F 14 1D 1F 11 13 18 14  FF
1790 : 16 12 16 11 0F 16 12 17  6F
1798 : 14 04 0C 04 07 02 04 05  CD
17A0 : 07 08 02 04 0C 08 0F 14  12
17A8 : 1B 1F 0A 13 18 18 2E 19  65
17B0 : 3F 3D 38 47 44 3A 2F A9  AB
17B8 : 00 85 06 85 07 A2 00 E4  FC
17C0 : E5 F0 0E 18 A5 06 69 50  75
17C8 : 85 06 90 02 E6 07 E8 D0  A1
17D0 : EE A5 06 18 65 E6 85 06  C5
17D8 : 90 02 E6 07 60 A2 12 A9  BB
17E0 : 00 20 CC CD E8 20 CC CD  3B
17E8 : 85 03 A9 03 20 26 18 A9  A1
17F0 : A7 85 04 60 20 DD 17 A0  F6
17F8 : 00 20 D8 CD A2 7F 20 AF  FE
1800 : 02 C8 D0 F5 E6 04 A5 04  86
1808 : C9 AF D0 EB 60 20 DD 17  07
1810 : A0 00 A2 7F 20 A2 02 20  A8
1818 : CA CD C8 D0 F5 E6 04 A5  07
```

Listing 1. »Window-Tech 80« wird mit dem MSE im C64-Modus eingegeben

Unschl

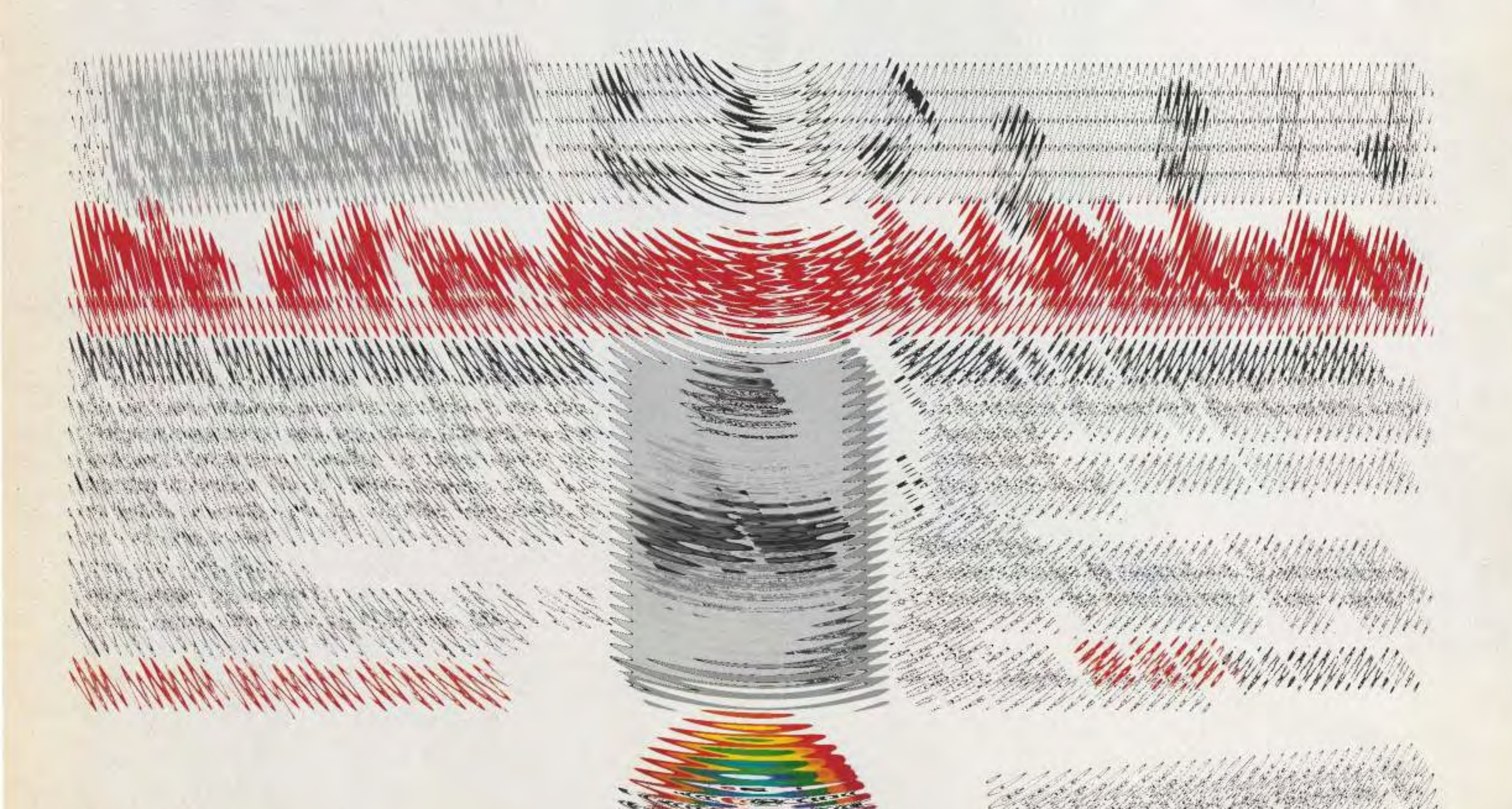





agbar!
64er online
64er-online.de
64er-online.net

```
1820 : 04 C9 AF D0 EB 60 8D B9  7A
1828 : 02 8D AA 02 60 38 3B 08  A0
1830 : 0B 10 13 18 1B 20 23 17  79
1838 : 20 B7 17 A2 12 A5 07 20  F9
1840 : CC CD E8 A5 06 4C 01 1A  DD
1848 : E6 E7 E6 E4 A5 E7 38 E5  BE
1850 : E6 85 08 85 09 E6 09 20  D8
1858 : 38 18 A0 00 20 D8 CD A2  0A
1860 : 7F 20 AF 02 C8 C4 08 D0  90
1868 : F3 A5 03 18 65 09 85 03  AC
1870 : 90 02 E6 04 E6 E5 A5 E5  3B
1878 : C5 E4 D0 DB 60 A5 E4 9D  61
1880 : 8F 17 A5 E5 9D 99 17 A5  0F
1888 : E6 9D A3 17 A5 E7 9D AD  74
1890 : 17 60 BD 8F 17 85 E4 BD  E5
1898 : 99 17 85 E5 BD A3 17 85  3B
18A0 : E6 BD AD 17 85 E7 60 07  DA
18A8 : AE A7 18 BD 71 17 85 04  D6
18B0 : A5 EB 9D 7B 17 A5 EC 9D  AF
18B8 : 85 17 20 7D 18 A9 00 85  5A
18C0 : 03 4C 48 18 AE A7 18 20  C7
18C8 : 92 18 BD 7B 17 85 EB BD  0E
18D0 : 85 17 85 EC 20 C4 1A 20  B1
18D8 : 53 19 20 3B 1A 20 53 19  49
18E0 : 20 14 1A AE A7 18 20 92  48
18E8 : 18 20 C4 1A 20 42 1A 4C  9A
18F0 : F5 18 EA EA EA AE A7 18  FC
18F8 : 20 92 18 E6 E7 E6 E4 A5  D9
1900 : E7 38 E5 E6 85 08 85 09  1A
1908 : E6 09 BD 71 17 85 04 A9  11
1910 : 00 85 03 20 38 18 A0 00  5F
1918 : A2 7F 20 A2 02 20 CA CD  BE
1920 : C8 C4 08 D0 F3 A5 03 18  0F
1928 : 65 09 85 03 90 02 E6 04  90
1930 : E6 E5 A5 E5 C5 E4 D0 DB  AD
1938 : AE A7 18 20 92 18 4C D2  85
1940 : 1A 11 91 60 60 A9 60 8D  43
1948 : 57 18 20 48 18 A9 20 8D  27
1950 : 57 18 60 A2 00 B5 E4 48  F2
1958 : E8 E0 04 D0 F8 20 45 19  A3
1960 : 20 B7 17 A5 07 18 69 10  CD
1968 : 85 07 20 3B 18 A0 00 A9  BA
1970 : 05 20 CA CD C8 C4 09 D0  6A
1978 : F8 E6 E5 A5 E5 C5 E4 D0  D3
1980 : DF A2 03 68 95 E4 CA 10  4A
1988 : FA 60 A9 00 8D 67 19 A9  E8
1990 : 66 8D 70 19 20 53 19 A9  50
1998 : 10 8D 67 19 A9 05 8D 70  46
19A0 : 19 60 C8 D0 FD E8 D0 FA  96
19A8 : A2 12 A9 08 20 CC CD E8  30
19B0 : A9 34 20 01 1A A9 BF 20  CA
19B8 : CA CD A5 D5 C9 58 D0 17  5E
19C0 : C8 EA EA D0 E3 AD B6 19  8B
19C8 : C9 BF D0 07 A9 A0 8D B6  C9
19D0 : 19 D0 D5 A9 BF D0 F7 A2  A3
19D8 : 00 DD 2D 18 F0 08 E8 E0  CA
19E0 : 0C D0 F6 60 EA EA E0 0B  BE
19E8 : F0 F9 86 1B 20 9E 1A AE  97
19F0 : A7 18 20 7D 18 20 A8 18  B0
19F8 : 20 1D 17 4C 09 1A 60 EA  AF
1A00 : EA 20 CC CD A9 03 4C 26  17
1A08 : 18 A6 1B 8E A7 18 20 C4  51
1A10 : 18 4C A2 1A 20 8A 19 4C  8E
1A18 : 04 1A C6 E5 10 02 E6 E5  10
1A20 : C6 E6 10 02 E6 E6 E6 E4  A9
1A28 : A5 E4 C9 19 D0 02 C6 E4  D7
1A30 : E6 E7 A5 E7 C9 50 90 02  D5
1A38 : C6 E7 60 E6 E5 E6 E6 4C  B0
1A40 : 26 1A 20 45 19 20 38 18  C8
1A48 : A0 00 A9 4F 20 CA CD C8  5E
1A50 : A9 63 20 CA CD C8 C4 08  52
1A58 : D0 F8 A9 50 20 CA CD C6  36
1A60 : E4 E6 E5 20 38 18 A0 00  FC
1A68 : A9 65 20 CA CD C8 A9 20  2F
1A70 : 20 CA CD C8 C4 08 D0 F8  43
1A78 : A9 67 20 CA CD E6 E5 A5  2D
1A80 : E5 C5 E4 D0 DE 20 38 18  9B
1A88 : A0 01 A9 4C 20 CA CD A9  80
1A90 : 64 20 CA CD C8 C4 08 D0  E5
1A98 : F8 A9 7A 4C CA CD 78 4C  23
1AA0 : F4 17 58 A9 58 85 EF EA  B2
1AA8 : EA C8 D0 FD E8 D0 FA 4C  84
1AB0 : BF 1A 38 E9 01 9D AD 17  A9
1AB8 : 60 E6 09 BD 71 17 60 A9  2A
1AC0 : 00 4C DA 1A 20 1A 1A C6  A9
1AC8 : E7 A5 E7 C9 4E D0 02 E6  F6
1AD0 : E7 60 20 17 FA 11 91 11  73
1AD8 : 00 60 85 D0 A2 12 A9 07  F3
1AE0 : 20 CC CD E8 A9 D0 20 CC  32
1AE8 : CD A0 00 A9 64 20 CA CD  49
1AF0 : C8 C0 10 D0 F8 AE A7 18  0A
1AF8 : E8 E0 0A D0 02 A2 00 8A  37
1B00 : 18 69 B0 48 A2 12 A9 08  73
1B08 : 20 CC CD E8 A9 34 20 CC  75
1B10 : CD 68 4C CA CD 18 24 38  1C
1B18 : A6 EC A5 EB 90 11 85 E4  8D
1B20 : 86 E7 4C 33 1B A5 ED A6  F7
1B28 : EE 20 1D CA A9 00 AA 85  17
1B30 : E5 86 E6 A9 00 A2 04 9D  A7
1B38 : 5D 03 CA D0 FA AE A7 18  D7
1B40 : 86 1B 78 20 7D 18 20 0D  A9
1B48 : 18 20 1D 17 20 09 1A 58  FE
1B50 : 20 7D FF 13 93 00 60 20  8C
1B58 : 7D FF 57 49 4E 44 4F 57  C7
1B60 : 2D 54 45 43 48 0D 4D 49  26
1B68 : 54 20 31 30 20 46 52 45  27
1B70 : 49 20 44 45 46 49 4E 49  FE
1B78 : 45 52 42 41 52 45 4E 20  68
1B80 : 46 45 4E 53 54 45 52 4E  BC
1B88 : 0D 5B 45 53 43 5D 20 22  E2
1B90 : 3F 22 20 3D 20 41 55 46  7E
1B98 : 52 55 46 0D 5B 45 53 43  7C
1BA0 : 5D 20 22 3C 22 20 3D 20  76
1BA8 : 44 45 46 20 4F 42 45 4E  DD
1BB0 : 2D 4C 49 4E 4B 53 0D 5B  5A
1BB8 : 45 53 43 5D 20 22 3E 22  74
1BC0 : 20 3D 20 44 45 46 20 55  C1
1BC8 : 4E 54 45 4E 20 52 45 43  8C
1BD0 : 48 54 53 0D 00 60 AD 00  73
1BD8 : FF 48 A2 02 8E 00 FF 20  0D
1BE0 : 50 14 20 CC 14 20 D5 15  A0
1BE8 : 20 00 17 20 57 1B 68 8D  DD
1BF0 : 00 FF 60 BD 42 BF 40 00  E3
```

Listing 1. (Schluß)



999 Grafikbilder im Speicher

Kaum zu glauben, aber »MSG« macht's möglich: Fantastische Grafikbilder auf komfortable Weise erstellen und bis zu 999 Bilder in eigene Programme einbinden.

Computerspiele werden durch ansprechende Grafik erst richtig interessant. Sowohl der C64 als auch der C128 sind hierfür sehr gut geeignet. Bei diesen Computern gibt es prinzipiell zwei Wege, Grafiken zu kreieren: Hires- oder Zeichensatzgrafik. Die erste Möglichkeit zeichnet sich durch große Flexiblität aus, belegt aber sehr viel Speicher im Computer, so daß bei einem Spiel nur wenig verschiedene Bilder verwendet werden können. Wenn man sich dagegen Spiele wie »Hexenküche« oder »Antiriad« anschaut, verfügen diese über eine große Anzahl von Bildern. Hier wurde die Zeichensatzgrafik verwendet. Doch dies allein reicht noch nicht aus. Betrachtet man sich die Bilder dieser Spiele genauer, so wird man feststellen, daß hier bestimmte Bildteile immer wieder auftauchen (Mauerteile, Plattformen, etc.). Um weiteren Speicherplatz zu sparen, werden solche Bildelemente als aus mehreren Zeichen bestehende Module gespeichert. Aus diesen setzt sich dann das endgültige Bild zusammen.



Bild 1. Solche fantastischen Bilder ermöglicht »MSG«

Der »Modul-Screen-Generator« (kurz MSG) macht diese Technik dem Hobbyprogrammierer zugänglich. Ein komfortabler Editor hilft beim Erstellen von Modulen, die dann zu Bildern kombiniert werden können. Fertige Bilder sind leicht in eigene Programme zu übernehmen. Welche faszinierenden Bilder Sie mit MSG trotz minimalem Speicherbedarf erzeugen können, sehen Sie in den Bildern 1 bis 3.

Module zum Platzsparen

Ein Modul des MSG besteht aus vier mal vier Zeichen, wobei jedes Zeichen eine eigene Farbe annehmen kann. In so einem Modul erstellt man kleine Grafikausschnitte, beispielsweise den Teil einer Mauer. Setzt man nun zweimal das gleiche Modul nebeneinander, wird die Mauer ein Stück größer. Wird dieser Vorgang nun beliebig oft wiederholt, lassen sich umfangreiche Bilder erzeugen. Innerhalb des MSGs dürfen bis zu 256 Module entworfen werden, wobei diese wiederum aus 256 selbstdefinierten Zeichen zusammengesetzt sind. Für diese kann entweder Hires- oder Multicolor verwendet werden. Aus den so erzeugten Modulen kann man schließlich im Screen-Editor eigene Bilder entwerfen.

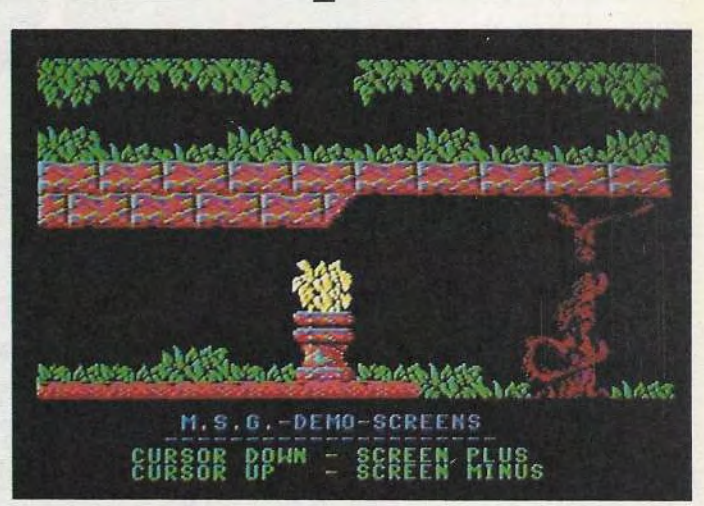


Bild 2. Faszinierende Bilder bei minimalem Speicherplatzbedarf können Sie in Ihre Programme einbinden

So arbeitet das Programm

Durch diese Methode lassen sich grafisch ansprechende Bilder erzielen. Aufgrund ihres geringen Platzbedarfs faßt der Speicher des C128 bis zu 999 (!) Stück davon.

Die erstellten Bilder und Module können problemlos in eigene Basic- oder Maschinenprogramme eingebaut werden, wobei bei Basic einige Abstriche gemacht werden müssen. Erstens funktionieren die Sprite- und Soundbefehle nicht mehr. Zweitens sollte man auf die Darstellung von hochauflösender Grafik verzichten – es sei denn, man legt den Zeichensatz an eine andere Speicherposition und schreibt den Bildschirmaufbau für seine Zwecke um. Drittens ist der Anzahl der Bilder bei der Programierung in Basic veringert. Alles in allem ermöglicht es MSG jedoch auch, Basic-Programme mit Hilfe der Grafiken ansprechend zu gestalten und aufzuwerten.

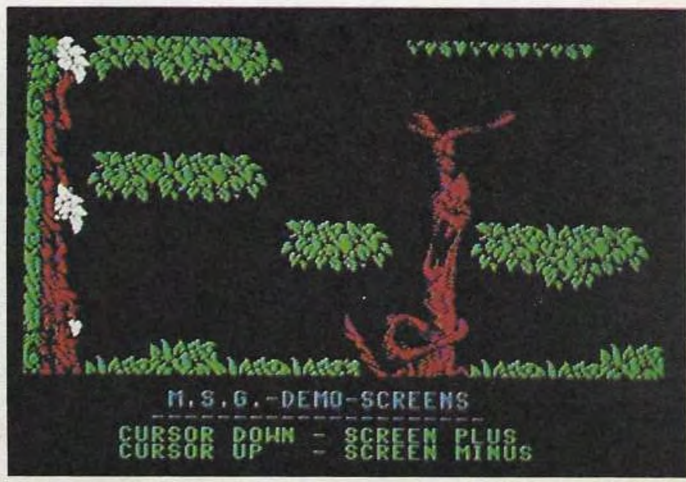


Bild 3. Der komfortable Editor macht die Erstellung schöner Bilder leicht und viele Ideen lassen sich realisieren



Der MSG besteht aus vier Teilen: dem Ladeprogramm, einigen Sprites, der Routine zum Bildschirmaufbau und dem eigentlichen Hauptprogramm. Es arbeitet im 40-Zeichen-Modus des C128. Bitte geben Sie Listing 1 bis 3 mit dem MSE im C64-Modus und Listing 4 (der Basic-Lader) mit dem Checksummer 128 ein. Beachten Sie dazu die Eingabehinweise auf Seite 158. Mit »RUN"M.S.G- V01.128"« wird der Lader (Listing 4) geladen und gestartet. Das Nachladen der anderen Programmteile erfolgt dann automatisch.

Editieren leicht gemacht

Beim Erstaufruf von MSG wird der Speicher von $3000 bis $FEFF in Bank 1 und das Modul Nummer 32 mit dem Space-Code beschrieben. Ferner wird das Zeichen mit dem Code 32 gelöscht. Dadurch ergibt ein Druck auf die Leertaste auch wirklich ein Leerzeichen.

Der MSG beinhaltet drei Editoren, die in zwei Gruppen aufgeteilt sind. Ein Programm dient dem Erstellen von Zeichensätzen und bildet mit dem Editor für Module eine Einheit. Ein weiteres Programm kümmert sich um das Zusammensetzen der Module zu Bildern und stellt die zweite Einheit dar.

Die Funktionen des Zeichensatzeditors:

TASTE	FUNKTION
<W>,<A>, <S>,<Z>	bewegt den Cursor über das Raster
<RETURN>	setzt einen Punkt an der Cursorposition
<SPACE>	löscht einen Punkt an der Cursorposition
<I>	invertiert das Raster
<HOME>	löscht das Raster
<↑>	scrollt den Inhalt des Zeichenrasters nach oben
<←>	scrollt den Inhalt des Zeichenrasters nach links
<X>	spiegelt Raster in X-Richtung
<Y>	spiegelt Raster in Y-Richtung
<HOME>	löscht Raster
<R>	holt das Zeichen unter dem Zeichensatzcursor ins Raster
<F5>	übernimmt das entworfene Zeichen in den aktuellen Zeichensatz. Der Zeichensatzcursor bestimmt dabei die Position.

Tabelle 1. So bedienen Sie den Zeichensatzeditor

Nach dem Start des Programms erscheint das erste Editorfeld vom MSG, welches Modul- und Zeichensatzeditor beinhaltet (Bild 4). In der linken unteren Ecke befindet sich ein acht mal acht Punkte großes Raster, das zum Entwerfen eigener Zeichen dient. Die Funktionen dieses Editors liegen in Tabelle 1 als Übersicht vor. Rechts von diesem Raster stehen einige Informationen, die das Arbeiten mit den Editoren erleichtern.

Infos erleichtern das Arbeiten

Da wäre an oberster Stelle die aktuelle Zeichennummer (»Z. NR.«), sie wird mit dem Cursor im Zeichensatzfeld rechts oben angewählt. Darunter steht die aktuelle Modulnummer (»M. NR.«), das dazugehörige Modul befindet sich in der Mitte ganz oben. Als nächstes folgt die Nummer des Zeichens, das sich gerade unter dem Cursor im Moduleditierfeld befindet (»A. NR.«). Schließlich wird noch über den aktuellen Zeichensatz (»Z. SATZ 0« oder »Z. Satz 1«) und die momentane Art der Zeichendarstellung Auskunft gegeben (»MULTI« oder »NORMAL«), wobei »MULTI«, Multicolor und »NORMAL« Hires entspricht. Des weiteren stehen rechts neben diesen Daten noch Informationen zu den aktuellen Farbwerten.

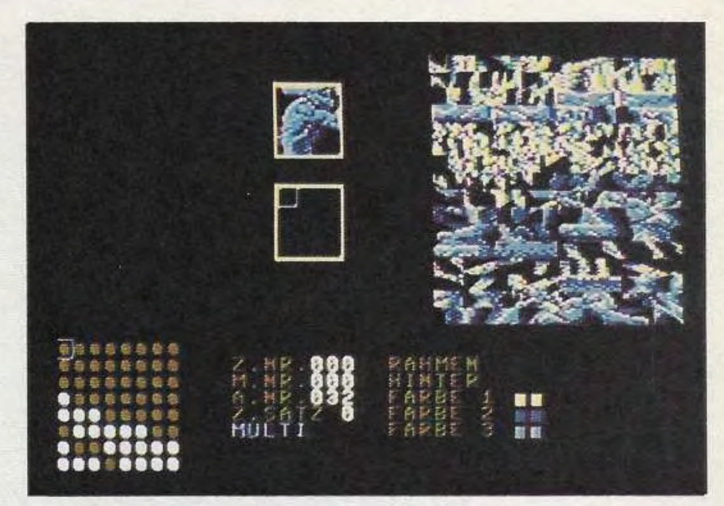


Bild 4. Der Modul- und Zeichensatzeditor

Doch nun zur Aufteilung des oberen Bildschirmbereichs. Hier findet ein Großteil des Moduleditors Platz, dessen Befehle Tabelle 2 als Übersicht zeigt.

Oben rechts befindet sich das Zeichensatzfeld (Bild 4). Hier werden die 256 Zeichen des gerade eingeschalteten Zeichensatzes abgebildet. Links daneben sieht man oben das aktuelle Modul, darunter das Moduleditierfeld. Um entworfene Module testen zu können, ist hierfür ein freier Platz auf der linken Bildschirmseite vorgesehen.

Soviel zum Aufbau des Modul- und Zeichensatzeditors. Um damit zu arbeiten, sollten Sie die Tabellen 1, 2 und 3 zu Rate ziehen: Hier finden sich die Befehle dieser zwei Programmbestandteile in einer Übersicht zusammengefaßt. Zur Beachtung: Wenn bei einzelnen Tasten in Klammern »10er Tast.« angegeben ist, so beziehen sich diese Angaben auf den abgesetzten Zehnerblock. Ebenso gilt bei »Cursortasten (Sonderblock)« der separate Cursorblock über dem normalen Tastenfeld.

Screen-Generierung leicht gemacht

Wir wollen uns nun dem zweiten Programmabschnitt zuwenden. Er wird mit einem Druck auf die Taste <F1> aufgerufen und dient dem Erstellen der Bilder. Dies geschieht durch Einsatz der zuvor definierten Module. Bild 5 zeigt diesen Programmteil.

Das Editorfeld teilt sich hier wie folgt auf: Die oberen 20 Zeilen sind reserviert für das Erstellen der Screens.

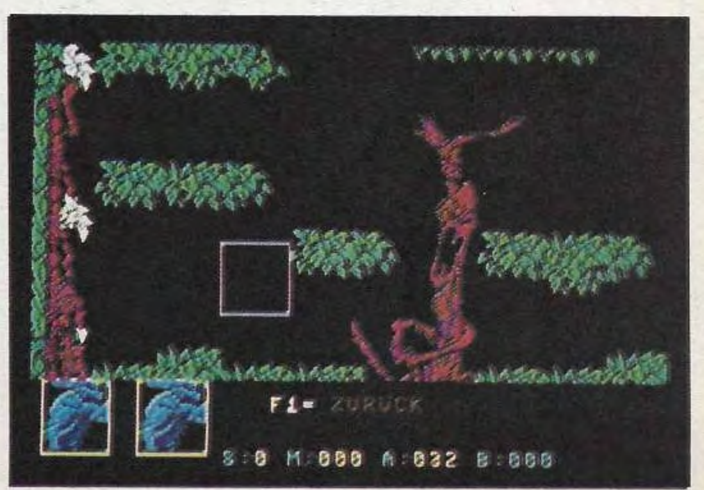


Bild 5. Mit dem Screeneditor können Sie grafisch anspruchsvolle Bilder aus Modulen zusammenstellen

MODULEDITOR Das Feld des Moduleditors sehen sie in der Mitte der oberen Bildschirmhälfte. Hier tritt ein zweiter Cursor auf.	
TASTE	**FUNKTION**
<CURSORTASTEN>	bewegen des Cursors im Moduleditorfeld (MEF)
<ENTER> (10er Tast.)	holt das Zeichen unter dem Zeichensatzfeld-Cursor an die Cursorposition im MEF und setzt Zeichenfarbe
<0> (10er Tast.)	setzt ein Space an die Cursorposition im MEF
<.> (10er Tast.)	füllt das MEF mit Spaces
<H>	holt aktuelles Modul in das MEF
<F7>	übernimmt das Modul aus dem MEF als aktuelles Modul
<SHIFT ←>	scrollt den Inhalt des MEF nach links
<SHIFT ↑>	scrollt den Inhalt des MEF nach oben
<F>	füllt das MEF mit dem Zeichen unter dem Zeichensatzfeldcursor
<SHIFT F>	füllt das MEF mit der aktuellen Zeichenfarbe

Tabelle 2. Die Befehle des Moduleditors. MEF ist hier die Abkürzung für Moduleditorfeld

Insgesamt finden hier 50 Module Platz, womit sich bei Grafiken ein sehr gutes Ergebnis erzielen läßt. In den unteren 5 Zeilen sieht man zwei umrahmte Felder. Das linke von beiden zeigt immer das aktuelle Modul aus dem sogenannten Modulspeicher an. Dieser Modulspeicher kann bis zu zehn Module aufnehmen. Er ist für öfter benötigte Module gedacht. Damit erspart man sich das ständige Herumblättern in der Modulbibliothek. Das rechte Feld zeigt immer das aktuelle Modul mit dem gerade gearbeitet wird. Die Nummer des Moduls aus dem Modul-Speicher sehen Sie am unterern Rand hinter der Bezeichnung »S:«. Das aktuelle Modul aus der Bibliothek wird bei »M:« angezeigt.

Zwei weitere Anzeigen (»A:« und »B:«) stehen für die Nummer des Moduls das sich gerade unter dem Cursor befindet, und für die aktuelle Bildnummer. Soviel zu den Anzeigen, die Befehle des Screeneditors sehen Sie in Tabelle 4.

Weitere wichtige Funktionen im ersten Editorteil:	
TASTE	**FUNKTION**
CURSORTASTEN (SONDERBLOCK)	bewegen den Cursor über das Zeichensatzfeld (Anzeige des Zeichencodes durch -»Z.NR.«)
<+> (10er Tast.)	schaltet Modulnummer um 1 weiter
<SHIFT +> (10er Tast.)	schaltet Modulnummer um 10 weiter
<-> (10er Tast.)	schaltet Modulnummer um 1 zurück
<SHIFT -> (10er Tast.)	schaltet Modulnummer um 10 zurück
<1>	erhöhen der Zeichenfarbe
<SHIFT 1>	erniedrigen der Zeichenfarbe
<2>	erhöhen der Muticolorfarbe 1
<SHIFT 2>	erniedrigen der Muticolorfarbe 1
<3>	erhöhen der Multicolorfarbe 2
<SHIFT 3>	erniedrigen der Multicolorfarbe 2
<4>	erhöhen der Hintergrundfarbe
<SHIFT 4>	erniedrigen der Hintergrundfarbe
<5>	erhöhen der Rahmenfarbe
<SHIFT 5>	erniedrigen der Rahmenfarbe
<M>	schaltet Multicolordarstellung ein
<N>	schaltet auf normale Zeichendarstellung
<T>	kopiert den Zeichensatz 0 nach Zeichensatz 1 (dieser wird dabei gelöscht)
<ESC>	wechselt den aktuellen Zeichensatz
<1> bis <9> (10er Tast.)	setzt das aktuelle Modul an den Platz im Testfeld, der der Position der Taste entspricht. (Beispiel: Taste 7 setzt das Modul links oben ins Testfeld)
<SHIFT 1> bis <SHIFT 9> (10er Tast.)	setzt an die jeweilige Stelle im Testfeld ein Space-Modul
<C>	füllt das Testfeld mit Space-Modulen
<F3>	springt ins Disk-Menü
<F1>	springt zum Screen-Editor

Tabelle 3. Weitere wichtige Funktionen im ersten Editorteil (für Zeichensätze und Module)

Diskettenfunktionen menügesteuert

Die beschriebenen Editoren dienen dem Erstellen von Bildern. Doch MSG kann noch mehr. Durch einen Druck auf die Taste <F3>, gelangt man aus beiden Edtitierteilen in das Disk-Menü. Dieser Teil des Programms bietet die Möglichkeit, die Module, die Bilder und den Zeichensatz zu speichern und zu laden. Man kann sich außerdem das Inhaltsverzeichnis einer Diskette anzeigen lassen, Befehle an die Floppy senden oder den Fehlerkanal auslesen. Bei der Ausgabe des Directories ist darauf zu achten, daß nur gescrollt wird, wenn die CONTROL-Taste gedrückt ist. Die Kommandos im Disk-Menü sind aus Tabelle 5 ersichtlich.

Zusammen mit den Editoren erhalten Sie die Möglichkeit, verschiedenste Bibliotheken aus Modulen anzulegen, zu speichern und wiederzuverwenden. So können Sie schöne Bilder sehr leicht erstellen. Die fertigen Produkte ihrer Kreativität können dann von eigenen Programmen aus aufgerufen werden. Darauf soll im folgenden eingegangen werden.

Bilder in eigenen Programmen

Bei der Verwendung von fertigen Bildern sind diese im Speicher abzulegen und bei Bedarf aufzurufen. MSG-Bilder können sowohl von Maschinen- als auch von Basic-Programmen verwaltet werden, wobei mit einem Assemblerprogramm wesentlich flexibler gearbeitet wird. Denn dabei kann man Bank 1 vollständig mit Daten für Module und Bilder belegen und den Speicher in Bank 0 für Steuerprogramm und sonstige Routinen verwenden.

Folgende Speicheraufteilung bietet sich besonders an:

BANK 0: Programm und Zeichensätze
(Aufteilung variabel)
BANK 1:
$1000-$2fff – Platz für 256 Module
$3000-$3fff – Platz für 999 Screens

Sollten Sie in Basic arbeiten, so hat dies einige Einschränkungen zur Folge. Listing 5 zeigt das Basic-Demo-Programm »DEMO .MSG«. Es muß mit dem Checksummer 128 eingegeben werden. Es lädt zu Beginn Zeichensatz, Module und Bilder sowie Maschinenroutinen nach. Natürlich müssen auf der Diskette ein vorher erstellter Zeichensatz, entsprechende Module und Bilder vorhanden sein. Dazu müssen auch deren Namen entsprechend in den Zeilen 140 bis 160 von Listing 5 eingetragen werden. Dann wird der Variablenzeiger nach oben gesetzt, um ein Überschreiben der Bild- und Moduldaten durch Variablen zu verhindern. Dies schränkt natürlich den zur Verfügung stehenden Speicher ein.

Im weiteren Verlauf initialisiert das Demo-Programm die Routine »BILDAUFBAU«, die sich um das Aufbauen der Bilder kümmert (Listing 6) und weiter »IRQ« zur Interruptsteuerung (Listing 7). Diese sind in Maschinensprache geschrieben und können gleichermaßen für Assembler-


64'ER ONLINE

SCREENEDITOR	
TASTE	**FUNKTION**
<CURSORTASTEN>	steuern den Cursor
<RETURN>	setzt das aktuelle Modul an Cursorposition
<SPACE>	setzt Space-Modul an Cursorposition
<R>	holt aktuellen Screen auf Bildschirm
<W>	übernimmt entworfenes Bild als aktuellen Screen
<HOME>	setzt Cursor in die linke obere Ecke
<CLR>	füllt Bildschirm mit Space-Modulen und setzt Cursor in die linke obere Ecke
<DEL>	setzt links vom Cursor ein Space-Modul, die restlichen Module werden nachgezogen
<INST>	fügt Space-Modul ein und setzt die Module rechts vom Cursor um 1 weiter nach rechts
<←>	sichert den sichtbaren Screen in einem Puffer
<SHIFT ←>	bringt den Screen aus Puffer auf den Bildschirm
<M>	schaltet Multicolordarstellung ein
<N>	schaltet auf normale Zeichendarstellung
<ESC>	wechselt den aktuellen Zeichensatz
<CURSORTASTEN> (SONDERBLOCK)	scrollt den Bildschirm um eine Modulreihe in die jeweilige Richtung
<.>(10er Tast.)	übernimmt das aktuelle Modul im aktuellen Modulspeicherplatz
<3>	erhöht Multicolorfarbe 2
<SHIFT 3>	erniedrigt Multicolorfarbe 2
<4>	erhöht Hintergrundfarbe
<SHIFT 4>	erniedrigt Hintergrundfarbe
<5>	erhöht Rahmenfarbe
<SHIFT 5>	erniedrigt Rahmenfarbe
<0> bis <9> (10er Tast.)	schaltet Modul mit der Nummer der jeweiligen Taste im Modulspeicher ein
<+>(10er Tast.)	erhöht Modulnummer um 1
<SHIFT +> (10er Tast.)	erhöht Modulnummer um 10
<->(10er Tast.)	erniedrigt Modulnummer um 1
<SHIFT -> (10er Tast.)	erniedrigt Modulnummer um 10
<+>	erhöht Bildnummer um 1
<SHIFT +>	erhöht Bildnummer um 10
<CBM +>	erhöht Bildnummer um 100
<->	erniedrigt Bildnummer um 1
<SHIFT ->	erniedrigt Bildnummer um 10
<CBM ->	erniedrigt Bildnummer um 100
<F3>	Sprung zum Disk-Menü
<F1>	zurück in den ersten Editorteil

Tabelle 4. Die Funktionen des Screeneditors erlauben ein komfortables und schnelles Erstellen von Grafiken.

und Basicprogramme verwendet werden. Über die Speicherstellen 3222/3223 dezimal ($0C96/0C97) wird der Routine für den Bildaufbau im Low/High-Byte-Format die Nummer des darzustellenden Bildes mitgeteilt.

Gestartet wird sie mit »SYS 2816«. Um also MSG-Bilder in eigenen Programmen unterzubringen, müssen die im Editor erstellten Komponenten (Zeichensatz, Module und Bilder) geladen und mit Hilfe der Routine in Listing 6 auf den Bildschirm gebracht werden.

Sollen mehr als zwei Zeichensätze beziehungsweise mehr als 999 Bilder in einem Programm verwendet werden, so können diese bei Bedarf von Programm nachgeladen werden.

Lassen Sie Ihrer Kreativität nun freien Lauf. Ihre schönsten Bilder inklusive der Zeichensätze können Sie uns ja unter dem Stichwort »MSG-128-Grafik« schicken.

(Arno Pichler/Axel Pretzsch)

DISK-MENÜ:	
TASTE	**FUNKTION**
<1>	Programmteile laden: Hier werden Zeichensätze, Module oder Bilder in den Speicher geholt. Der Zeichensatz läßt sich wahlweise an Position eins oder zwei laden. Module werden an einen beliebigen Platz in der Modulbibliothek geladen, wobei darauf zu achten ist, daß der Modulspeicher ausreicht. Das gilt auch für Bilder.
<2>	Programmteile speichern. Arbeitet analog zur Ladefunktion. Bilder und Module speichert man durch Eingabe von Anfangs- und Endnummer auch in Paketen. Unsinnige Eingaben führen zum Hauptmenü (beispielsweise bei Modulnummer größer als 255).
<3>	Directory anzeigen. Gibt das Inhaltsverzeichis der aktuellen Disk auf dem Bildschirm aus. Beachten Sie bitte, daß nur bei Drücken von <CONTROL> gescrollt wird.
<SPACE>	sorgt für Rückkehr ins Hauptmenü.
<4>	Diskbefehl senden.
<5>	Fehlerkanal auslesen.
<6>	Diskmodus verlassen. Sprung in den vor Aufruf des Disk-Menü aktuellen Editorteil

Tabelle 5. Die Kommandos im Disk-Menü. Das Disk-Menü können Sie aus beiden Editoren mit <F3> anwählen.


```
Name : msg.1                   3000 455b
-----------------------------------------
3000 : a9 05 8d 06 d5 ad 46 45  be
3008 : f0 06 20 85 0d 8d 46 45  95
3010 : 20 3e 38 a9 b2 8d ed 39  54
3018 : 20 25 3c 20 1d 37 20 82  ef
3020 : 39 20 c4 33 a0 04 a9 07  e0
3028 : 99 27 d0 88 10 f8 a2 38  5e
3030 : 8e f8 07 8e f9 07 e8 8e  67
3038 : fa 07 8e fb 07 8e fc 07  c0
3040 : a9 03 8d 17 d0 8d 1d d0  41
3048 : a9 00 8d 1c d0 ad 3a 45  c6
3050 : 8d 10 d0 a0 09 b9 b5 37  d1
3058 : 99 00 d0 88 10 f7 a9 12  c2
3060 : a2 0a 85 eb 86 ec 20 14  5f
3068 : ca 20 42 c1 20 7d ff 95  24
3070 : 20 5a 2e 4e 52 2e 05 30  1e
3078 : 30 30 20 20 95 52 41 48  4e
3080 : 4d 45 4e 20 20 90 12 cf  76
3088 : cf 92 0d 95 20 4d 2e 4e  58
3090 : 52 2e 05 30 30 30 20 20  86
3098 : 95 48 49 4e 54 45 52 20  66
30a0 : 20 90 12 cf cf 92 0d 95  78
30a8 : 20 41 2e 4e 52 2e 05 30  c9
30b0 : 30 30 20 20 95 46 41 52  3a
30b8 : 42 45 20 31 20 9e 12 cf  aa
30c0 : cf 92 0d 95 20 5a 2e 53  03
30c8 : 41 54 5a 20 05 30 20 20  61
30d0 : 95 46 41 52 42 45 20 32  56
30d8 : 20 05 12 cf cf 92 0d 9a  f4
30e0 : 20 4e 4f 52 4d 41 4c 20  96
30e8 : 20 20 20 95 46 41 52 42  0f
30f0 : 45 20 33 20 1c 12 cf cf  47
30f8 : 92 00 ad 47 0d c9 18 d0  00
3100 : 0d a9 30 8d 5a 07 a9 20  64
3108 : 8d 75 39 4c 18 31 a9 31  3c
3110 : 8d 5a 07 a9 28 8d 75 39  f9
3118 : 20 e6 34 20 13 39 20 09  4a
3120 : 34 20 1c 32 20 c5 37 a2  04
3128 : 04 8e 37 45 20 bf 34 ae  18
3130 : 37 45 ca 10 f4 20 13 38  cb
3138 : a9 1f 8d 15 d0 ee 29 d0  42
3140 : ee 2a d0 ee 2b d0 20 2b  65
3148 : 45 a5 d4 c9 58 f0 ee a0  d8
3150 : 00 d9 6a 31 f0 08 c8 c0  f2
3158 : 32 d0 f6 4c 3d 31 98 0a  0d
3160 : a8 b9 9d 31 48 b9 9c 31  99
3168 : 48 60 04 05 53 54 55 56  5c
3170 : 07 02 09 0a 0d 0c 51 4c  0b
3178 : 3c 01 47 44 4f 45 42 4d  52
3180 : 46 41 4e 33 52 14 17 19  b5
3188 : 39 36 49 4a 38 3b 08 0b  0c
3190 : 10 21 06 03 48 11 1d 15  bf
3198 : 16 24 27 27 d2 3f fd 39  01
31a0 : fd 31 2d 32 47 32 6e 32  ec
31a8 : 96 32 cc 32 01 33 30 33  a2
31b0 : 47 33 18 33 6f 33 6b 33  a2
31b8 : da 33 d7 33 d3 36 d6 36  3f
31c0 : d9 36 dc 36 df 36 e2 36  5a
31c8 : e5 36 e8 36 eb 36 9b 33  0e
31d0 : bd 33 50 34 1f 35 3d 35  bc
31d8 : 84 35 ac 35 fb 33 3a 34  73
31e0 : a7 34 aa 34 ad 34 b0 34  7a
31e8 : b3 34 76 34 16 35 5f 33  c8
31f0 : 3f 37 36 37 8e 37 79 36  34
31f8 : ab 36 f9 37 fc 37 ad f5  50
3200 : 39 c9 10 90 11 38 e9 10  cf
3208 : 8d f5 39 ad 07 d0 e9 08  43
3210 : 8d 07 d0 8d bc 37 20 1c  45
3218 : 32 4c 3d 31 ad f5 39 8d  70
3220 : f0 39 a9 00 8d f1 39 a2  aa
3228 : e0 a0 06 4c e8 37 ad f5  4e
```

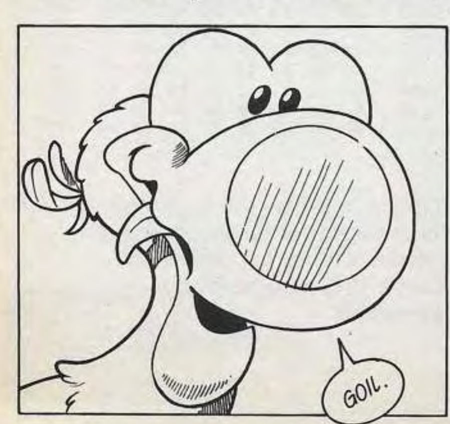
```
3230 : 39 c9 f0 b0 e1 18 69 10   45
3238 : 8d f5 39 ad 07 d0 69 08   71
3240 : 8d 07 d0 8d bc 37 d0 ce   9d
3248 : ad ac 37 f0 c9 ce ac 37   6b
3250 : ad 06 d0 38 e9 08 8d 06   5d
3258 : d0 8d bb 37 b0 0b ad 10   ff
3260 : d0 49 08 8d 10 d0 8d 3a   bb
3268 : 45 ce f5 39 4c 16 32 ad   52
3270 : ac 37 c9 0f f0 f6 ee ac   e8
3278 : 37 18 ad 06 d0 69 08 8d   7b
3280 : 06 d0 8d bb 37 90 0b ad   49
3288 : 10 d0 49 08 8d 10 d0 8d   0b
3290 : 3a 45 ee f5 39 d0 d5 a5   a4
3298 : d3 29 01 d0 19 ad a8 37   6a
32a0 : c9 03 f0 0f ee a8 37 ad   75
32a8 : 09 d0 18 69 08 8d be 37   a3
32b0 : 8d 09 d0 4c bf 37 ad a8   3d
32b8 : 37 f0 f8 ce a8 37 ad 09   8c
32c0 : d0 38 e9 08 8d 09 d0 8d   a7
32c8 : be 37 4c bf 37 a5 d3 29   6f
32d0 : 01 d0 18 ad a9 37 c9 03   77
32d8 : f0 d9 ee a9 37 18 ad 08   a1
32e0 : d0 69 08 8d 08 d0 8d bd   d1
32e8 : 37 d0 c8 ad a9 37 f0 c3   0f
32f0 : ce a9 37 38 ad 08 d0 e9   9a
32f8 : 08 8d 08 d0 8d bd 37 4c   1f
3300 : bf 37 ad aa 37 f0 0f ce   f0
3308 : aa 37 38 ad 05 d0 e9 08   a0
3310 : 8d 05 d0 8d ba 37 4c 3d   17
3318 : 31 ad aa 37 c9 07 f0 f6   38
3320 : ee aa 37 18 ad 05 d0 69   4d
3328 : 08 8d 05 d0 8d ba 37 d0   7f
3330 : e5 ad ab 37 f0 e0 ce ab   66
3338 : 37 38 ad 04 d0 e9 08 8d   0f
3340 : 04 d0 8d b9 37 4c 3d 31   74
3348 : ad ab 37 c9 07 f0 c7 ee   c7
3350 : ab 37 18 ad 04 d0 69 08   cf
3358 : 8d 04 d0 8d b9 37 d0 b6   d3
3360 : 20 ae 0c 20 5f 37 20 09   21
3368 : 34 4c 3d 31 ad f5 39 2c   00
3370 : a9 20 85 fa ad a8 37 0a   fb
3378 : aa bd ad 37 85 fb 85 fd   9d
3380 : bd ae 37 85 fc 18 69 d4   f3
3388 : 85 fe ac a9 37 a5 fa 91   9c
3390 : fb ad 42 0d 91 fd 20 d4   c7
3398 : 37 4c 3d 31 a0 07 a9 09   66
33a0 : 99 a8 da 99 d0 da 99 f8   b3
33a8 : da 99 20 db 99 48 db 99   51
33b0 : 70 db 99 98 db 99 c0 db   cc
33b8 : 88 10 e5 4c 3d 31 20 c4   b3
33c0 : 33 4c bf 37 a0 03 a9 20   f9
33c8 : 99 4e 05 99 76 05 99 9e   30
33d0 : 05 99 c6 05 88 10 f1 60   86
33d8 : a9 01 2c a9 09 85 fd 20   37
33e0 : 24 ca ad aa 37 18 69 11   26
33e8 : 85 fe a6 fe ac ab 37 18   ab
33f0 : 20 f0 ff a4 ec a5 fd 91   34
33f8 : e2 4c 3d 31 a5 d3 d0 2f   10
3400 : ee ad 0c 20 09 34 4c 3d   aa
3408 : 31 ad ad 0c 8d f0 39 a9   95
3410 : 00 8d f1 39 a2 08 a0 07   75
3418 : 20 e8 37 a9 5e a2 04 85   c5
3420 : fb 86 fc a9 5e a2 d8 85   3c
3428 : 8e 86 8f 20 8a 0c 60 18   9c
3430 : ad ad 0c 69 0a 8d ad 0c   c0
3438 : 4c 03 34 a5 d3 d0 06 ce   41
3440 : ad 0c 4c 03 34 38 ad ad   7e
3448 : 0c e9 0a 8d ad 0c 4c 03   f0
3450 : 34 a9 00 a2 04 85 fb 86   17
3458 : fc a2 0c a0 00 a9 20 91   ad
3460 : fb c8 c0 0c d0 f9 a9 28   45
3468 : 18 65 fb 85 fb 90 02 e6   fc
3470 : fc ca d0 e7 4c 3d 31 a9   c9
3478 : a8 a2 da 85 fb 86 fc a2   06
3480 : 07 a0 00 b1 fb 29 0f c9   e6
3488 : 01 f0 03 a9 01 2c a9 09   22
3490 : 91 fb c8 c0 08 d0 ec a9   77
3498 : 28 18 65 fb 85 fb 90 02   24
34a0 : e6 fc ca 10 dc 4c 3d 31   40
34a8 : a2 00 2c a2 01 2c a2 02   aa
34b0 : 2c a2 03 2c a2 04 20 d9   f2
34b8 : 34 20 bf 34 4c 3d 31 8a   fb
34c0 : 0a a8 b9 0d 35 85 fb b9   11
34c8 : 0e 35 85 fc a0 00 bd 49   05
34d0 : 0d 91 fb c8 91 fb 4c 13   0e
34d8 : 39 a5 d3 d0 06 fe 49 0d   8a
34e0 : 4c e6 34 de 49 0d 20 01   08
34e8 : 35 ad 48 0d c9 c8 f0 10   6e
34f0 : ad 49 0d 09 08 8d 42 0d   b6
34f8 : ad 49 0d 29 07 8d 49 0d   ce
3500 : 60 a0 04 b9 49 0d 99 42   d1
3508 : 0d 88 10 f7 60 3d db 65   86
3510 : db 8d db 15 db ed da 20   24
3518 : ae 39 20 5f 37 4c 3d 31   84
3520 : 20 5b 35 a2 07 a0 07 b1   84
3528 : fb 48 88 10 fa a0 07 68   0d
3530 : 91 fb 88 10 fa 20 6c 35   b0
3538 : ca 10 ea 4c 3d 31 20 5b   e3
3540 : 35 a2 03 a0 07 b1 fb 48   1a
3548 : b1 fd 91 fb 68 91 fd 88   f8
3550 : 10 f3 20 6c 35 ca 10 eb   b1
3558 : 4c 3d 31 a9 a8 a2 da 85   da
3560 : fb 86 fc a9 c0 a2 db 85   ae
3568 : fd 86 fe 60 a9 28 18 65   7b
3570 : fb 85 fb 90 02 e6 fc 38   fa
3578 : a5 fd e9 28 85 fd a5 fe   78
3580 : e9 00 85 fe 60 a5 d3 f0   0f
3588 : 03 4c f0 35 20 5b 35 a2   8b
3590 : 07 a0 00 b1 fb 48 c8 b1   a6
3598 : fb 88 91 fb c8 c0 07 d0   0b
35a0 : f5 68 91 fb 20 6c 35 ca   7d
35a8 : 10 e7 4c 3d 31 a5 d3 f0   d8
35b0 : 03 4c 3e 36 20 5b 35 a9   35
35b8 : d0 a2 da 85 fd 86 fe a0   92
35c0 : 07 b1 fb 48 88 10 fa a2   e2
35c8 : 06 a0 07 b1 fd 91 fb 88   84
35d0 : 10 f9 20 6c 35 a9 50 18   85
35d8 : 65 fd 85 fd 90 02 e6 fe   10
35e0 : ca 10 e6 a0 00 68 91 fb   01
35e8 : c8 c0 08 d0 f8 4c 3d 31   76
35f0 : a9 4e a2 05 85 fb 86 fc   56
35f8 : a9 4e a2 d9 85 fd 86 fe   0c
3600 : a2 03 a0 00 b1 fb 48 b1   cb
3608 : fd 48 c8 c0 04 d0 f5 a0   53
3610 : 02 68 91 fd 68 91 fb 88   7e
3618 : 10 f7 a0 03 68 91 fd 68   88
3620 : 91 fb a9 28 18 65 fb 85   c6
3628 : fb 90 02 e6 fc a9 28 18   b7
3630 : 65 fd 85 fd 90 02 e6 fe   68
3638 : ca 10 c7 4c bf 37 a2 03   cc
3640 : bd 4e 05 48 bd 4e d9 48   b5
3648 : bd 76 05 9d 4e 05 bd 76   26
3650 : d9 9d 4e d9 bd 9e 05 9d   e7
3658 : 76 05 bd 9e d9 9d 76 d9   ac
3660 : bd c6 05 9d 9e 05 bd c6   0c
3668 : d9 9d 9e d9 68 9d c6 d9   35
3670 : 68 9d c6 05 ca 10 c9 4c   e6
3678 : bf 37 a5 d3 f0 17 a2 03   0f
3680 : ad 42 0d 9d 4e d9 9d 76   5c
3688 : d9 9d 9e d9 9d c6 d9 ca   20
3690 : 10 f1 4c 3d 31 a2 03 ad   e3
3698 : f5 39 9d 4e 05 9d 76 05   7c
36a0 : 9d 9e 05 9d c6 05 ca 10   61
36a8 : f1 4c bf 37 a9 00 a2 20   fb
36b0 : 85 fb 86 fc a9 00 a2 28   e9
36b8 : 85 fd 86 fe a2 08 a0 00   aa
36c0 : b1 fb 91 fd c8 d0 f9 e6   5c
36c8 : fc e6 fe ca d0 f2 20 5f   34
36d0 : 37 4c 3d 31 a2 28 2c a2   04
36d8 : 2a 2c a2 2c 2c a2 14 2c   c7
36e0 : a2 16 2c a2 18 2c a2 00   5a
36e8 : 2c a2 02 2c a2 04 bd 26   f9
36f0 : 0c 85 fb 85 8e bd 27 0c   fa
36f8 : 85 fc 18 69 d4 85 8f a5   32
3700 : d3 d0 06 20 8a 0c 4c 3d   76
3708 : 31 a2 03 a0 03 a9 20 91   80
3710 : fb 88 10 fb 20 6c 35 ca   a3
3718 : 10 f1 4c 3d 31 a9 a8 a2   24
3720 : 06 85 fb 86 fc a2 07 a0   fb
3728 : 07 a9 51 91 fb 88 10 fb   c7
3730 : 20 6c 35 ca 10 f1 60 20   7f
3738 : 82 39 20 5f 37 4c 3d 31   78
3740 : 20 4e 37 ad 5a 07 49 01   10
3748 : 8d 5a 07 4c 3d 31 ad 47   f0
3750 : 0d 49 02 8d 47 0d ad 75   b3
3758 : 39 49 08 8d 75 39 60 a9   e0
3760 : 0f 8d 18 d4 a9 00 8d 06   b3
3768 : d4 8d 04 d4 8d 00 d4 8d   e6
3770 : 05 d4 a9 2c 8d 01 d4 20   44
3778 : 2b 45 a9 fa 8d 06 d4 a9   bf
3780 : 02 8d 05 d4 a9 15 8d 04   a6
3788 : d4 a9 14 8d 04 d4 60 a9   a3
3790 : 4e a2 05 85 fb 86 fc a9   5d
3798 : 4e a2 d9 85 8e 86 8f 20   fa
37a0 : 8a 0c 20 5f 37 4c bf 37   68
37a8 : 00 00 00 00 00 4e 05 76   1c
37b0 : 05 9e 05 c6 05 86 40 86   b1
37b8 : 70 17 b9 d7 31 88 71 20   7a
37c0 : c5 37 4c 3d 31 ad a8 37   6d
37c8 : 0a aa bd ad 37 85 fb bd   57
37d0 : ae 37 85 fc ac a9 37 b1   73
37d8 : fb 8d f0 39 a9 00 8d f1   b2
37e0 : 39 a2 30 a0 07 20 e8 37   0e
37e8 : 86 fb 84 fc 20 c4 38 a0   77
37f0 : 02 b9 f2 39 91 fb 88 10   ee
37f8 : f8 60 a9 d8 2c a9 c8 8d   f4
3800 : 48 0d 20 13 38 20 e6 34   c2
3808 : a2 00 20 bf 34 20 13 39   ad
3810 : 4c 3d 31 ad 48 0d c9 c8   a2
3818 : d0 0c a2 05 bd 38 38 9d   f1
3820 : 7b 07 ca 10 f7 60 a2 05   eb
3828 : bd 32 38 9d 7b 07 ca 10   fb
3830 : f7 60 0d 15 0c 14 09 20   03
3838 : 0e 0f 12 0d 01 0c 78 a9   9a
3840 : 5d 8d 14 03 a9 38 8d 15   86
3848 : 03 a9 01 8d 12 d0 ad 11   92
3850 : d0 29 7f 8d 11 d0 a9 81   88
3858 : 8d 1a d0 58 60 d8 20 6a   54
3860 : 38 20 87 fc 20 e7 c6 4c   1f
3868 : 33 ff ad 19 d0 8d 19 d0   a9
3870 : 29 01 d0 01 60 ad 12 d0   cb
3878 : cd ed 39 b0 2a ad 46 0d   e3
3880 : 8d 20 d0 ad 45 0d 8d 21   3c
3888 : d0 ad 48 0d 8d 16 d0 ad   0b
3890 : 43 0d 8d 22 d0 ad 44 0d   a7
3898 : 8d 23 d0 ad 47 0d 8d 18   e4
38a0 : d0 ad ed 39 4c c0 38 a9   e8
38a8 : 15 8d 18 d0 a9 00 8d 20   b5
38b0 : d0 8d 21 d0 a9 c8 8d 16   ec
38b8 : d0 a9 15 8d 18 d0 a9 01   05
38c0 : 8d 12 d0 60 ad f0 39 8d   f9
38c8 : ee 39 ad f1 39 8d ef 39   2f
38d0 : a9 30 8d f2 39 8d f3 39   95
38d8 : 8d f4 39 ad ee 39 d0 08   ef
38e0 : ad ef 39 f0 2d ce ef 39   6d
38e8 : ce ee 39 ad f4 39 c9 39   e4
38f0 : f0 05 ee f4 39 d0 e4 a9   be
38f8 : 30 8d f4 39 ad f3 39 c9   46
3900 : 39 f0 05 ee f3 39 d0 d3   c4
3908 : a9 30 8d f3 39 ee f2 39   f4
3910 : d0 c9 60 a9 18 85 fb 85   bb
3918 : fd a9 04 85 fc a9 d8 85   27
3920 : fe a9 00 85 fa a2 10 a0   ea
3928 : 00 a5 fa 91 fb ad 42 0d   3c
3930 : 91 fd e6 fa c8 c0 10 d0   4d
3938 : f0 a9 28 18 65 fb 85 fb   4e
3940 : 90 02 e6 fc a9 28 18 65   31
3948 : fd 85 fd 90 02 e6 fe ca   82
3950 : d0 d5 60 a9 00 8d 8b 39   65
3958 : ad f5 39 8d 8a 39 0e 8a   c0
3960 : 39 2e 8b 39 0e 8a 39 2e   31
3968 : 8b 39 0e 8a 39 2e 8b 39   0a
3970 : 18 ad 8b 39 69 20 8d 8b   4e
3978 : 39 a9 a8 a2 da 85 fd 86   e3
3980 : fe 60 20 53 39 a2 00 a0   0b
3988 : 00 bd 00 20 39 f6 39 d0   3d
3990 : 03 a9 09 2c a9 01 91 fd   15
3998 : c8 c0 08 d0 ec a9 28 18   c9
39a0 : 65 fd 85 fd 90 02 e6 fe   d8
39a8 : e8 e0 08 d0 da 60 20 53   f4
39b0 : 39 ad 8a 39 8d da 39 ad   79
39b8 : 8b 39 8d db 39 a2 00 a0   a9
39c0 : 00 84 fb b1 fd 29 0f c9   30
39c8 : 09 f0 07 a5 fb 19 f6 39   97
39d0 : 85 fb c8 c0 08 d0 ec a5   a3
39d8 : fb 9d 00 20 a9 28 18 65   ad
39e0 : fd 85 fd 90 02 e6 fe e8   56
39e8 : e0 08 d0 d3 60 b2 00 00   16
39f0 : 00 00 00 00 00 00 80 40   73
39f8 : 20 10 08 04 02 01 a9 13   98
3a00 : a2 30 4c 09 3a a9 d3 a2   74
3a08 : 3f 8d 23 3c 8e 24 3c 20   99
3a10 : 25 3c a9 01 8d ed 39 a9   5e
3a18 : 00 85 9d 8d 15 d0 20 68   1d
3a20 : 3f 20 7d ff 93 9f 12 20   8d
3a28 : 20 4d 2e 53 2e 47 2e 20   fb
3a30 : 44 49 53 4b 2d 4d 45 4e   46
3a38 : 55 45 20 20 92 0d 11 11   34
3a40 : 11 11 05 31 2e 95 20 50   f2
3a48 : 47 52 2e 54 45 49 4c 45   29
3a50 : 20 4c 41 44 45 4e 0d 0d   84
3a58 : 05 32 2e 95 20 50 47 52   fb
3a60 : 2e 54 45 49 4c 45 20 53   49
3a68 : 50 45 49 43 48 45 52 4e   aa
3a70 : 0d 0d 05 33 2e 95 20 44   44
3a78 : 49 52 45 43 54 4f 52 59   60
3a80 : 20 5a 45 49 47 45 4e 0d   3a
3a88 : 0d 05 34 2e 95 20 44 49   e9
3a90 : 53 4b 2d 42 45 46 45 48   49
3a98 : 4c 20 53 45 4e 44 45 4e   2b
3aa0 : 0d 0d 05 35 2e 95 20 44   b4
3aa8 : 49 53 4b 2d 53 54 41 54   99
```

Listing 1. MSG (Teil 1), der komfortable Editor für Zeichensatzgrafiken. Bitte mit dem MSE (Seite 158) im C64-Modus eingeben.

```
3ab0 : 55 53 0d 0d 05 36 2e 95  7a
3ab8 : 20 4d 45 4e 55 45 20 56  47
3ac0 : 45 52 4c 41 53 53 45 4e  eb
3ac8 : 00 a9 00 85 d0 20 e4 ff  ef
3ad0 : f0 fb c9 31 90 f7 c9 37  b5
3ad8 : b0 f3 38 e9 31 0a a8 b9  47
3ae0 : e9 3a 48 b9 e8 3a 48 60  72
3ae8 : 2a 3c f8 3c f3 3a 91 3b  c4
3af0 : ca 3b 1e 3c 20 25 3c a9  d6
3af8 : 03 a2 05 85 eb 86 ec 20  25
3b00 : 14 ca a9 24 8d 80 0a 20  ad
3b08 : 7d ff 9e 13 00 20 cc ff  c3
3b10 : a9 00 85 62 8d 00 ff 38  b0
3b18 : 20 c3 ff a9 01 a2 80 a0  b7
3b20 : 0a 20 bd ff a9 00 a2 08  df
3b28 : a0 60 20 ba ff 20 c0 ff  5c
3b30 : b0 4c a2 00 20 c6 ff 20  27
3b38 : b4 b8 a0 03 84 63 20 cf  54
3b40 : ff 85 60 a5 90 d0 37 20  7b
3b48 : cf ff 85 61 a5 90 d0 2e  23
3b50 : c6 63 d0 ea 20 07 ba a9  d2
3b58 : 00 a2 08 a0 03 20 5d ba  dc
3b60 : a9 20 20 d2 ff 20 cf ff  bc
3b68 : f0 09 a6 90 d0 10 20 d2  4c
3b70 : ff 90 f2 20 b4 b8 a5 d3  c7
3b78 : f0 fc a0 02 d0 be 20 cc  6c
3b80 : ff a9 00 38 20 c3 ff 20  bb
3b88 : f2 0c 20 e4 ff f0 fb 4c  35
3b90 : 0f 3a a9 0f 20 c3 ff 20  69
3b98 : 09 3c 20 7d ff 95 42 45  b7
3ba0 : 46 45 48 4c 20 3a 20 9e  b6
3ba8 : 00 a9 0f a8 a2 08 20 ba  b6
3bb0 : ff a9 00 20 bd ff 20 c0  66
3bb8 : ff a2 0f 20 c9 ff 20 cf  8d
3bc0 : ff 20 d2 ff c9 0d d0 f6  ba
3bc8 : 20 cc ff 20 09 3c 20 7d  40
3bd0 : ff 12 9e 00 a9 01 a2 08  bd
3bd8 : a0 0f 20 ba ff a9 00 20  ed
3be0 : bd ff 20 c0 ff a2 01 20  16
3be8 : c6 ff 20 cf ff 20 d2 ff  fc
3bf0 : 24 90 50 f6 20 cc ff a9  0b
3bf8 : 01 20 c3 ff 20 e4 ff f0  05
3c00 : fb a9 92 20 d2 ff 4c 0f  f5
3c08 : 3a 20 5f 37 20 24 ca a2  a5
3c10 : 14 a0 05 18 20 f0 ff 20  82
3c18 : 52 ca a9 08 85 ec 60 20  bc
3c20 : f2 0c 4c 3d 31 20 24 ca  0d
3c28 : 4c 42 c1 20 25 3c 20 68  8f
3c30 : 3f 20 4f 3e b9 3e 3c 48  2a
3c38 : b9 3d 3c 48 60 42 3c 8a  c6
3c40 : 3c c7 3c 20 95 3f 85 fb  d4
3c48 : 86 fc a9 00 a2 70 8d cf  3a
3c50 : 3f 8e d0 3f 20 af 3e a9  be
3c58 : 3f aa 20 68 ff 20 79 3c  61
3c60 : ae cf 3f ac d0 3f a9 00  09
3c68 : 20 d5 ff a5 90 f0 07 a5  1f
3c70 : fc f0 03 20 73 3f 4c cb  a3
3c78 : 3b a9 02 a2 08 a0 00 20  23
3c80 : ba ff a5 b7 a2 80 a0 0a  5f
3c88 : 4c bd ff 20 09 3c 20 7d  a5
3c90 : ff 95 41 42 20 4d 4f 44  25
3c98 : 55 4c 20 4e 52 2e 05 30  f0
3ca0 : 30 30 9d 9d 9d 00 a2 20  a8
3ca8 : a0 10 20 d3 3e 8e cf 3f  e9
3cb0 : 8c d0 3f c0 30 90 03 4c  b8
3cb8 : 0f 3a 20 af 3e a9 00 85  1f
3cc0 : fc a9 01 a2 00 4c 5a 3c  6a
3cc8 : 20 09 3c 20 7d ff 95 41  31
3cd0 : 42 20 42 49 4c 44 20 4e  e0
3cd8 : 52 2e 05 30 30 30 9d 9d  bf
3ce0 : 9d 00 a2 35 a0 30 20 d3  80
3ce8 : 3e 8e cf 3f 8c d0 3f c0  17
3cf0 : fe 90 c7 e0 9f b0 c0 90  e8
3cf8 : c1 20 25 3c 20 68 3f 20  1d
3d00 : 4f 3e b9 0c 3d 48 b9 0b  71
3d08 : 3d 48 60 10 3d 49 3d e0  58
3d10 : 3d 20 95 3f 8d cf 3f 8e  1c
3d18 : d0 3f 18 69 00 8d d1 3f  ed
3d20 : 8a 69 08 8d d2 3f 20 af  1a
3d28 : 3e a9 0e aa 20 68 ff 20  99
3d30 : 79 3c ad cf 3f 85 fb ad  98
3d38 : d0 3f 85 fc a9 fb ae d1  82
3d40 : 3f ac d2 3f 20 d8 ff 4c  d3
3d48 : cb 3b 20 09 3c 20 7d ff  95
3d50 : 95 56 4f 4e 20 4d 4f 44  e0
3d58 : 55 4c 20 4e 52 2e 3a 05  2f
3d60 : 30 30 30 9d 9d 9d 00 a2  74
3d68 : 20 a0 10 20 d3 3e 8e cf  e9
3d70 : 3f 8c d0 3f c0 2f 90 06  e5
3d78 : d0 d0 e0 e1 b0 cc 20 09  29
3d80 : 3c 20 7d ff 95 42 49 53  63
3d88 : 20 4d 4f 44 55 4c 20 4e  80
3d90 : 52 2e 3a 05 30 30 30 9d  a9
3d98 : 9d 9d 00 a2 20 a0 10 20  e0
3da0 : d3 3e 8e d1 3f 8c d2 3f  92
3da8 : c0 2f 90 09 f0 03 4c 0f  bc
3db0 : 3a e0 e1 b0 f9 a9 20 18  86
3db8 : 6d d1 3f 8d d1 3f 90 03  ef
3dc0 : ee d2 3f ad d2 3f cd d0  9d
3dc8 : 3f 90 e3 d0 0a ad d1 3f  36
3dd0 : cd cf 3f 90 d9 f0 d7 20  2b
3dd8 : af 3e a9 01 a2 00 4c 2c  e5
3de0 : 3d 20 09 3c 20 7d ff 95  10
3de8 : 56 4f 4e 20 42 49 4c 44  a6
3df0 : 20 4e 52 2e 3a 05 30 30  7f
3df8 : 30 9d 9d 9d 00 a2 [illegible] [illegible]  3d
3e00 : 30 20 d3 3e 8e cf 3f 8c  7a
3e08 : d0 3f c0 fe 90 06 d0 d1  a8
3e10 : e0 9f b0 cd 20 09 3c 20  21
3e18 : 7d ff 95 42 49 53 20 42  77
3e20 : 49 4c 44 20 4e 52 2e 3a  49
3e28 : 05 30 30 30 9d 9d 9d 00  95
3e30 : a2 35 a0 30 20 d3 3e 8e  52
3e38 : d1 3f 8c d2 3f c0 fe 90  3d
3e40 : 06 d0 09 e0 9f b0 05 a9  f4
3e48 : 35 4c b7 3d 4c 0f 3a 20  9f
3e50 : 7d ff 93 9f 12 20 20 4d  e3
3e58 : 2e 53 2e 47 2e 20 44 49  2c
3e60 : 53 4b 2d 4d 45 4e 55 45  f4
3e68 : 20 20 92 0d 11 11 11 11  df
3e70 : 11 05 31 2e 20 95 5a 45  b9
3e78 : 49 43 48 45 4e 53 41 54  4b
3e80 : 5a 0d 0d 05 32 2e 20 95  85
3e88 : 4d 4f 44 55 4c 45 0d 0d  76
3e90 : 05 33 2e 20 95 42 49 4c  e8
3e98 : 44 45 52 0d 0d 00 20 e4  d0
3ea0 : ff c9 31 90 f9 c9 34 b0  02
3ea8 : f5 38 e9 31 0a a8 60 20  02
3eb0 : 09 3c 20 7d ff 95 4e 41  f7
3eb8 : 4d 45 20 3a 20 9e 00 a2  33
3ec0 : 00 20 cf ff c9 0d f0 08  9d
3ec8 : 9d 80 0a e8 e0 10 d0 f1  fb
3ed0 : 86 b7 60 8e cd 3f 8c ce  c2
3ed8 : 3f a4 ec 8c cc 3f a6 eb  6f
3ee0 : 8e cb 3f a2 00 20 cf ff  b8
3ee8 : c9 0d f0 10 c9 30 90 f5  c2
3ef0 : c9 3a b0 f1 9d 80 0a e8  18
3ef8 : e0 03 d0 e9 ac cc 3f ae  57
3f00 : cb 3f 18 20 f0 ff a2 00  0f
3f08 : 86 83 86 84 a4 ec b1 e0  bc
3f10 : c9 30 f0 10 a9 64 18 65  18
3f18 : 83 85 83 90 02 e6 84 20  fa
3f20 : 60 3f d0 ea c8 b1 e0 c9  e2
3f28 : 30 f0 10 a9 0a 18 65 83  08
3f30 : 85 83 90 02 e6 84 20 60  af
3f38 : 3f d0 ea c8 b1 e0 c9 30  5d
3f40 : f0 0b e6 83 d0 02 e6 84  a2
3f48 : 20 60 3f d0 ef ad cd 3f  a4
3f50 : 20 f9 0b a5 83 18 69 00  83
3f58 : aa a5 84 6d ce 3f a8 60  ee
3f60 : 38 b1 e0 e9 01 91 e0 60  c7
3f68 : a9 01 a2 08 85 eb 86 ec  e7
3f70 : 4c 14 ca a9 00 a2 70 85  90
3f78 : fd 86 fe a9 3f 8d 00 ff  0d
3f80 : a2 08 a0 00 b1 fd 91 fb  98
3f88 : c8 d0 f9 e6 fc e6 fe ca  ac
3f90 : d0 f2 4c f2 0c 20 09 3c  a9
3f98 : 20 7d ff 95 5a 45 49 43  a5
3fa0 : 48 45 53 41 54 5a 20 05  2a
3fa8 : 12 31 92 95 4f 44 45 52  7b
3fb0 : 20 12 05 32 92 00 20 e4  d4
3fb8 : ff c9 31 f0 09 c9 32 d0  50
3fc0 : f5 a9 00 a2 28 60 a9 00  0a
3fc8 : a2 20 60 00 00 00 00 00  93
3fd0 : 00 00 00 a9 f1 8d ed 39  bb
3fd8 : 20 25 3c a2 05 bd ed 40  65
3fe0 : 9d f8 07 bd f3 40 9d 27  79
3fe8 : d0 ca 10 f1 a2 0b bd f9  cd
3ff0 : 40 9d 00 d0 ca 10 f7 a9  79
3ff8 : 07 8d 17 d0 8d 1d d0 a9  fe
4000 : 3f 8d 15 d0 ad 3b 45 8d  4a
4008 : 10 d0 a9 06 8d 1b d0 a2  e6
4010 : 18 a0 0c 18 20 f0 ff 20  48
4018 : 7d ff 9f 53 3a 9e 30 20  81
4020 : 9f 4d 3a 9e 30 30 30 20  4e
4028 : 9f 41 3a 9e 30 30 30 20  50
4030 : 9f 42 3a 30 30 30 13 00  56
4038 : 20 7f 44 20 a5 44 20 d5  d6
```

Listing 1. (Fortsetzung)

ROCKUS








64ER ONLINE

```
4040 : 44 4c 47 40 20 6a 43 20  27
4048 : 00 0b a0 04 b9 42 0d 99  8c
4050 : 49 0d 88 10 f7 ee 27 d0  79
4058 : 20 2b 45 a5 d4 c9 58 f0  f3
4060 : f4 a2 00 dd 83 40 f0 08  6f
4068 : e8 e0 28 d0 f6 4c 55 40  8c
4070 : 8a c9 0a 90 35 38 e9 0a  44
4078 : 0a aa bd b4 40 48 bd b3  82
4080 : 40 48 60 51 47 44 4f 45  85
4088 : 42 4d 46 41 4e 53 54 55  a6
4090 : 56 07 02 04 3c 01 11 09  8d
4098 : 28 2b 49 4a 4c 52 33 00  16
40a0 : 39 40 48 05 3b 08 0b 10  ec
40a8 : 24 27 8d 36 45 20 a5 44  fe
40b0 : 4c 55 40 7f 41 aa 41 04  1d
40b8 : 42 d5 41 04 41 3b 41 12  cd
40c0 : 30 35 43 2f 43 60 43 c6  13
40c8 : 43 88 42 c5 42 00 43 19  fc
40d0 : 43 cf 43 e0 43 55 42 15  fa
40d8 : 44 f2 43 35 42 4c 42 04  a4
40e0 : 3a 0b 45 0e 45 11 45 14  cd
40e8 : 45 22 45 1f 45 38 38 38  db
40f0 : 3a 3b 3c 07 07 07 01 09  77
40f8 : 09 16 30 1e d1 4e d1 90  d4
4100 : da ae da c6 da a5 d3 d0  8c
4108 : 1c ad 39 45 c9 04 b0 0f  8f
4110 : ad 01 d0 18 69 20 8d 01  45
4118 : d0 8d fa 40 ee 39 45 20  83
4120 : f5 44 4c 55 40 ad 39 45  d6
4128 : f0 f5 ad 01 d0 38 e9 20  55
4130 : 8d 01 d0 8d fa 40 ce 39  83
4138 : 45 4c 1f 41 a5 d3 d0 21  12
4140 : ad 38 45 c9 09 b0 d8 18  3e
4148 : ad 00 d0 69 20 8d 00 d0  67
4150 : 8d f9 40 90 06 ee 10 d0  b6
4158 : ee 3b 45 ee 38 45 4c 1f  30
4160 : 41 ad 38 45 f0 b9 38 ad  48
4168 : 00 d0 e9 20 8d 00 d0 8d  86
4170 : f9 40 b0 06 ce 10 d0 ce  c4
4178 : 3b 45 ce 38 45 4c 1f 41  c6
4180 : a2 09 bd 0d 0d 48 bd 17  f0
4188 : 0d 9d 0d 0d bd 21 0d 9d  9d
4190 : 17 0d bd 2b 0d 9d 21 0d  5f
4198 : bd 35 0d 9d 2b 0d 68 9d  df
41a0 : 35 0d ca 10 dd 20 f5 44  50
41a8 : 4c 44 40 a2 09 bd 35 0d  e8
41b0 : 48 bd 2b 0d 9d 35 0d bd  77
41b8 : 21 0d 9d 2b 0d bd 17 0d  62
41c0 : 9d 21 0d bd 0d 0d 9d 17  c7
41c8 : 0d 68 9d 0d 0d ca 10 dd  35
41d0 : 20 f5 44 4c 44 40 a9 0d  8c
41d8 : a2 0d 85 fb 86 fc a2 05  c6
41e0 : a0 09 b1 fb 48 88 b1 fb  78
41e8 : c8 91 fb 88 d0 f7 68 91  1a
41f0 : fb a9 0a 18 65 fb 85 fb  8a
41f8 : 90 02 e6 fc ca d0 e1 20  dd
4200 : f5 44 4c 44 40 a9 0d a2  7e
4208 : 0d 85 fb 86 fc a2 05 a0  e2
4210 : 00 b1 fb 48 c8 b1 fb 88  0c
4218 : 91 fb c8 c0 09 d0 f5 68  b1
4220 : 91 fb a9 0a 18 65 fb 85  02
4228 : fb 90 02 e6 fc ca d0 df  f2
4230 : 20 f5 44 4c 44 40 20 47  3b
4238 : 43 a6 fb bd 0d 0d ac 36  dd
4240 : 45 99 51 45 20 a5 44 20  cf
4248 : 5f 37 4c 55 40 20 4e 37  ad
4250 : 20 5f 37 4c 55 40 a5 d3  0d
4258 : f0 03 20 7e 42 a9 00 8d  2e
4260 : 38 45 8d 39 45 8d 10 d0  68
4268 : 8d 3b 45 a9 16 8d 00 d0  89
4270 : 8d f9 40 a9 30 8d 01 d0  54
4278 : 8d fa 40 4c 44 40 a9 20  49
4280 : a0 31 99 0d 0d 88 10 fa  0c
4288 : 60 a5 d3 c9 01 f0 07 c9  30
4290 : 02 f0 06 a2 00 2c a2 09  de
4298 : 2c a2 63 20 a7 42 ca 10  ca
42a0 : fa 20 d5 44 4c 47 40 ee  86
42a8 : 24 0c d0 03 ee 25 0c ad  0a
42b0 : 25 0c c9 03 90 0f ad 24  2f
42b8 : 0c c9 e7 90 08 a9 00 8d  9e
42c0 : 24 0c 8d 25 0c 60 a5 d3  f4
42c8 : c9 01 f0 07 c9 02 f0 06  ab
42d0 : a2 00 2c a2 09 2c a2 63  15
42d8 : 20 e4 42 ca 10 fa 20 d5  59
42e0 : 44 4c 47 40 ad 24 0c d0  f2
42e8 : 03 ce 25 0c ce 24 0c ad  b7
42f0 : 25 0c c9 ff 90 0a a9 03  94
42f8 : 8d 25 0c a9 e6 8d 24 0c  d4
4300 : 60 a5 d3 d0 09 ee ad 0c  19
4308 : 20 7f 44 4c 55 40 ad ad  ec
4310 : 0c 18 69 0a 8d ad 0c 4c  d3
4318 : 08 43 a5 d3 d0 06 ce ad  79
4320 : 0c 4c 08 43 ad ad 0c 38  a6
4328 : e9 0a 8d ad 0c 4c 08 43  f9
4330 : ae ad 0c 4c 38 43 a2 20  aa
4338 : 20 47 43 a4 fb 8a 99 0d  f6
4340 : 0d 20 f5 44 4c 44 40 ad  a6
4348 : 38 45 85 fb ad 39 45 85  c8
4350 : fc a5 fc f0 0b c6 fc 18  87
4358 : a9 0a 65 fb 85 fb d0 f1  3e
4360 : 60 20 d5 44 20 5f 37 4c  41
4368 : 47 40 ad 43 0d 29 0f 0a  0e
4370 : 0a 0a 0a 8d 41 0d ad 44  6f
4378 : 0d 29 0f 0d 41 0d 8d 41  b5
4380 : 0d ad 45 0d 29 0f 0a 0a  9e
4388 : 0a 0a 8d 3f 0d ad 46 0d  54
4390 : 29 0f 0d 3f 0d 8d 3f 0d  c0
4398 : ad 47 0d c9 18 f0 07 ad  e6
43a0 : 40 0d 09 01 d0 05 ad 40  36
43a8 : 0d 29 fe 8d 40 0d ad 48  6f
43b0 : 0d c9 d8 d0 07 ad 40 0d  eb
43b8 : 09 02 d0 05 ad 40 0d 29  fa
43c0 : fd 09 80 8d 40 0d 60 20  42
43c8 : fa 0c 20 5f 37 4c 44 40  24
43d0 : ae 36 45 20 47 43 a4 fb  08
43d8 : bd 51 45 99 0d 0d 4c 44  b5
43e0 : 40 ae 36 45 ad ad 0c 9d  61
43e8 : 51 45 20 a5 44 20 5f 37  ca
43f0 : 4c 55 40 a5 d3 d0 11 a0  f5
43f8 : 34 b9 0d 0d 99 50 0d 88  4f
4400 : 10 f7 20 5f 37 4c 55 40  ac
4408 : a0 34 b9 50 0d 99 0d 0d  27
4410 : 88 10 f7 4c 47 40 a5 d3  dc
4418 : d0 1f 20 6f 44 ac 38 45  83
4420 : c0 09 f0 0b c8 b1 fb 88  1d
4428 : 91 fb c8 c0 09 d0 f5 a9  43
4430 : 20 91 fb 20 f5 44 4c 44  57
4438 : 40 20 6f 44 a0 09 b1 fb  fe
4440 : c9 20 d0 28 ac 38 45 c0  76
4448 : 09 f0 21 b1 fb 99 47 45  7c
4450 : c8 c0 09 d0 f6 ac 38 45  15
4458 : b9 47 45 c8 91 fb c0 09  2d
4460 : d0 f6 ac 38 45 a9 20 91  23
4468 : fb 20 f5 44 4c 44 40 ad  bc
4470 : 39 45 0a aa bd 3c 45 85  02
4478 : fb bd 3d 45 85 fc 60 ad  67
4480 : ad 0c 8d f0 39 a9 00 8d  b1
4488 : f1 39 a2 d2 a0 07 20 e8  ad
4490 : 37 a9 27 a2 07 85 fb 86  54
4498 : fc a9 27 a2 db 85 8e 86  b8
44a0 : 8f 20 8a 0c 60 ad ad 0c  a6
44a8 : 8d 37 45 ae 36 45 bd 51  1f
44b0 : 45 8d ad 0c 18 8a 69 30  85
44b8 : 8d ce 07 a9 21 a2 07 85  f2
44c0 : fb 86 fc a9 21 a2 db 85  14
44c8 : 8e 86 8f 20 8a 0c ad 37  af
44d0 : 45 8d ad 0c 60 ad 24 0c  e5
44d8 : 8d f0 39 ad 25 0c 8d f1  ae
44e0 : 39 a2 de a0 07 20 e8 37  b9
44e8 : 20 7e 0b ad 40 0d 29 80  d2
44f0 : d0 03 20 6a 43 20 47 43  70
44f8 : a6 fb bd 0d 0d 8d f0 39  20
4500 : a9 00 8d f1 39 a2 d8 a0  98
4508 : 07 4c e8 37 a2 01 2c a2  7e
4510 : 02 2c a2 03 2c a2 04 20  5a
4518 : d9 34 20 2b 45 4c 44 40  c1
4520 : a9 c8 2c a9 d8 8d 48 0d  a3
4528 : 4c 44 40 a0 38 a2 00 ca  e9
4530 : d0 fd 88 d0 fa 60 00 00  ee
4538 : 00 00 00 00 0d 0d 17 0d  e8
4540 : 21 0d 2b 0d 35 0d ff 00  10
4548 : 00 00 00 00 00 00 00 00  49
4550 : 00 00 00 00 00 00 00 00  51
4558 : 00 00 00 00 ff 00 ff 10  79
```

Listing 1. (Schluß)

```
Name : msg.2                 0b00 0db6
--------------------------------------
0b00 : ad ad 0c 85 fd a9 00 8d  80
0b08 : f0 39 ae f0 39 bd 0d 0d  2e
0b10 : 8d ad 0c 8a 0a aa bd 26  01
0b18 : 0c 85 fb 85 8e bd 27 0c  22
0b20 : 85 fc 18 69 d4 85 8f 20  4f
0b28 : 8a 0c ee f0 39 ad f0 39  c9
0b30 : c9 32 d0 d6 ad 3f 0d 48  bb
0b38 : 29 0f 8d 46 0d 68 6a 6a  a8
0b40 : 6a 6a 29 0f 8d 45 0d ad  9e
0b48 : 40 0d 29 01 d0 03 a9 18  75
0b50 : 2c a9 1a 8d 47 0d ad 40  9d
0b58 : 0d 29 02 d0 03 a9 c8 2c  8d
0b60 : a9 d8 8d 48 0d ad 41 0d  3f
0b68 : 48 29 0f 8d 44 0d 68 6a  dd
0b70 : 6a 6a 6a 29 0f 8d 43 0d  54
0b78 : a5 fd 8d ad 0c 60 8d 02  33
0b80 : ff 20 b7 0b a0 34 b1 fd  4d
0b88 : 99 0d 0d 88 10 f8 4c f2  dc
0b90 : 0c a9 04 18 65 87 85 87  2d
0b98 : 90 02 e6 88 a9 04 18 65  da
0ba0 : 89 85 89 90 02 e6 8a 18  12
0ba8 : a5 fb 69 28 85 fb 85 8e  16
0bb0 : 90 04 e6 fc e6 8f 60 ad  63
0bb8 : 24 0c 85 83 ad 25 0c 85  f3
0bc0 : 84 a9 35 20 f9 0b 18 a5  0e
0bc8 : 83 69 00 85 fd a5 84 69  a2
0bd0 : 30 85 fe 60 85 83 a9 00  aa
0bd8 : 85 84 a9 20 20 f9 0b 18  3c
0be0 : a5 83 69 00 85 87 a5 84  d5
0be8 : 69 10 85 88 a5 87 69 10  28
0bf0 : 85 89 a5 88 69 00 85 8a  76
0bf8 : 60 85 85 a2 00 86 86 86  2c
0c00 : 8b 86 8c a0 10 d0 0d 18  f1
0c08 : a5 8b 65 85 85 8b a5 8c  e1
0c10 : 65 86 85 8c 46 8c 66 8b  25
0c18 : 66 84 66 83 88 30 04 90  06
0c20 : f3 b0 e4 60 00 00 00 04  b9
0c28 : 04 04 08 04 0c 04 10 04  da
0c30 : 14 04 18 04 1c 04 20 04  37
0c38 : 24 04 a0 04 a4 04 a8 04  1c
0c40 : ac 04 b0 04 b4 04 b8 04  f1
0c48 : bc 04 c0 04 c4 04 40 05  2e
0c50 : 44 05 48 05 4c 05 50 05  02
0c58 : 54 05 58 05 5c 05 60 05  5f
0c60 : 64 05 e0 05 e4 05 e8 05  44
0c68 : ec 05 f0 05 f4 05 f8 05  19
0c70 : fc 05 00 06 04 06 80 06  2e
0c78 : 84 06 88 06 8c 06 90 06  29
0c80 : 94 06 98 06 9c 06 a0 06  87
0c88 : a4 06 20 ea 0c ad ad 0c  92
0c90 : 20 d4 0b a2 03 a0 00 b1  ca
0c98 : 87 91 fb b1 89 91 8e c8  0e
0ca0 : c0 04 d0 f3 20 91 0b ca  65
0ca8 : 10 eb 4c f2 0c 00 ad ad  f2
0cb0 : 0c 20 d4 0b a9 4e a2 05  04
0cb8 : 85 fb 86 fc a9 4e a2 d9  c7
0cc0 : 85 8e 86 8f a2 03 a0 00  e5
0cc8 : b1 fb 20 ea 0c 91 87 20  88
0cd0 : f2 0c b1 8e 29 0f 20 ea  68
0cd8 : 0c 91 89 20 f2 0c c8 c0  47
0ce0 : 04 d0 e5 20 91 0b ca 10  86
0ce8 : dd 60 48 a9 7e 8d 00 ff  91
0cf0 : 68 60 48 a9 0e 8d 00 ff  1d
0cf8 : 68 60 8d 02 ff 20 b7 0b  2a
0d00 : a0 34 b9 0d 0d 91 fd 88  31
0d08 : 10 f8 4c f2 0c 00 00 00  c7
0d10 : 00 00 00 00 00 00 00 00  11
0d18 : 00 00 00 00 00 00 00 00  19
0d20 : 00 00 00 00 00 00 00 00  21
0d28 : 00 00 00 00 00 00 00 00  29
0d30 : 00 00 00 00 00 00 00 00  31
0d38 : 00 00 00 00 00 00 00 00  39
0d40 : 00 00 07 02 06 00 00 18  d3
0d48 : c8 07 02 06 00 00 18 c8  c7
0d50 : 00 00 00 00 00 00 00 00  51
0d58 : 00 00 00 00 00 00 00 00  59
0d60 : 00 00 00 00 00 00 00 00  61
0d68 : 00 00 00 00 00 00 00 00  69
0d70 : 00 00 00 00 00 00 00 00  71
0d78 : 00 00 00 00 00 00 00 00  79
0d80 : 00 00 00 00 00 8d 02 ff  f5
0d88 : a9 00 a2 30 85 fb 86 fc  2c
0d90 : a2 cf a0 00 a9 20 91 fb  1c
0d98 : c8 d0 fb e6 fc ca d0 f4  f7
0da0 : a0 1f 99 00 14 88 10 fa  f2
0da8 : 20 f2 0c a9 00 a0 07 99  ce
0db0 : 00 21 88 10 fa 60 85 20  6e
```

Listing 2. Der MSG, Teil 2 (bitte mit dem MSE (Seite 158) eingeben)

```
Name : msg.3                0e00 0f40
-------------------------------------
0e00 : ff ff c0 80 00 40 80 00   43
0e08 : 40 80 00 40 80 00 40 80   9b
0e10 : 00 40 80 00 40 80 00 40   d9
0e18 : 80 00 40 80 00 40 80 00   bd
0e20 : 40 80 00 40 80 00 40 80   b3
0e28 : 00 40 80 00 40 80 00 40   f1
0e30 : 80 00 40 ff ff c0 00 00   c6
0e38 : 00 00 00 00 00 00 00 00   39
0e40 : ff c0 00 80 40 00 80 40   36
0e48 : 00 80 40 00 80 40 00 80   a4
0e50 : 40 00 80 40 00 80 40 00   be
0e58 : 80 40 00 ff c0 00 00 00   04
0e60 : 00 00 00 00 00 00 00 00   61
0e68 : 00 00 00 00 00 00 00 00   69
0e70 : 00 00 00 00 00 00 00 00   71
0e78 : 00 00 00 00 00 00 00 00   79
0e80 : fc 30 00 c0 30 00 c0 70   93
0e88 : fc f0 30 00 c0 30 fc c0   0b
0e90 : 30 00 c0 fc 00 00 00 00   90
0e98 : 00 00 00 00 00 00 00 00   99
0ea0 : 00 00 00 00 00 00 00 00   a1
0ea8 : 00 00 00 00 00 00 00 00   a9
0eb0 : 00 00 00 00 00 00 00 00   b1
0eb8 : 00 00 00 00 00 00 00 00   b9
0ec0 : fc cc f8 0c cc cc 18 cc   0f
0ec8 : cc 30 cc f8 60 cc f0 c0   b0
0ed0 : cc d8 fc 78 cc 00 00 00   23
0ed8 : 00 00 00 00 00 00 00 00   d9
0ee0 : 00 00 00 00 00 00 00 00   e1
0ee8 : 00 00 00 00 00 00 00 00   e9
0ef0 : 00 00 00 00 00 00 00 00   f1
0ef8 : 00 00 00 00 00 00 00 00   f9
0f00 : cc 78 cc 00 cc d8 cc c0   84
0f08 : f0 cc c0 e0 cc c0 f0 cc   db
0f10 : cc d8 78 78 cc 00 00 00   42
0f18 : 00 00 00 00 00 00 00 00   19
0f20 : 00 00 00 00 00 00 00 00   21
0f28 : 00 00 00 00 00 00 00 00   29
0f30 : 00 00 00 00 00 00 00 00   31
0f38 : 00 00 00 00 00 00 00 00   39
```

Listing 3. Der MSG, Teil 3 (bitte mit dem MSE eingeben)

```
10 POKE 53280,0: POKE 53281,0                              <RD1>
20 SCNCLR                                                  <VRJ>
30 CHAR ,3,7,"{LIG.BLUE}ΩPYP{7SPACE,YELLOW}ΩY{7SP
   ACE}": PRINT CHR$(159);" ΩY"                            <URL>
40 CHAR ,3,8,"{LIG.BLUE}Π{2SPACE}Ψ{WHITE} ODUL-{Y
   ELLOW} YP{WHITE} CREEN-": PRINT CHR$(159);" ΠP
   {WHITE} ENERATOR"                                       <6GG>
50 CHAR ,13,9,"{YELLOW}YYY{6SPACE}": PRINT CHR$(1
   59);"{2SPACE}YY"                                        <EOO>
60 CHAR ,13,11,"{YELLOW,2SPACE}1987 BY"                    <3UA>
70 CHAR ,13,13,"ARNO PICHLER"                              <JUN>
80 CHAR ,9,14,"FRITZ-SCHAEFFER-STR.8"                      <9BP>
90 CHAR ,10,15,"8201 OSTERMUENCHEN"                        <V9V>
100 BLOAD "MSG.2"                                          <8VE>
110 BLOAD "MSG.3"                                          <CNF>
120 BANK 12: BOOT "MSG.1"                                  <G9I>
```

Listing 4. »M.S.G.-V01.128«, der Lader des MSG. bitte mit dem Checksummer 128 eingeben.

```
10 REM ***********************************                 <A99>
20 REM *                                 *                 <AHA>
30 REM *    M.S.G.- DEMO-PROGRAMM        *                 <APB>
40 REM *                                 *                 <B1C>
50 REM *        (C) 1987 BY              *                 <B9D>
60 REM *                                 *                 <BHE>
70 REM *        ARNO PICHLER             *                 <BPF>
80 REM *                                 *                 <810>
90 REM ***********************************                 <891>
100 :                                                      <2C9>
110 :                                                      <2SB>
120 GRAPHIC 1: GRAPHIC 0: REM BASIC-ANFANG NACH $
    4000                                                   <M1S>
130 BANK 1 : REM BANK 12 EINSCHALTEN                       <VVA>
140 BLOAD "ZEICHENSATZ",ON B0 : REM ZEICHENSATZ N
    ACH $2000 IN BANK 0 LADEN                              <7L0>
150 BLOAD "MODULE",ON B1 : REM MODULE NACH $1000
    IN BANK 1 LADEN                                        <VP6>
160 BLOAD "BILDER",ON B1 : REM BILDDATEN NACH $30
    00 IN BANK 1 LADEN                                     <HRU>
170 BLOAD "IRQ",ON B0 : REM IRQ-ROUTINE NACH $130
    0 IN BANK 0                                            <9VQ>
180 BLOAD "BILDAUFBAU",ON B0 : REM PRG.ZUM BILDAU
    FBAU NACH $0B00 IN BANK 0                              <JM2>
190 POKE 47,152: POKE 48,58: CLR : REM VARIABLEN-
    ANFANG NACH OBEN SETZEN                                <K6P>
200 BI=0 : REM BILDNUMMER AUF NULL SETZEN                  <QF3>
210 BN=3222 : REM ADRESSE FUER BILDNUMMER                  <NCB>
220 IQ=4998 : REM PROGRAMMEINSPRUNG IRQ-ALT                <TT3>
230 IR=4864 : REM PROGRAMMEINSPRUNG IRQ-NEU                <06L>
240 BA=2816 : REM PROGRAMMEINSPRUNG BILDSCHIRM AU
    FBAUEN                                                 <6Q9>
250 SCNCLR : REM BILDSCHIRM LOESCHEN                       <KJR>
260 SYS IR : REM NEUER IRQ EIN                             <VF9>
270 CHAR ,9,21,"{LIG.BLUE}M.S.G.-DEMO-SCREENS"             <EPV>
280 CHAR ,8,22,"---------------------"                     <OKT>
290 CHAR ,6,23,"{GREEN}CURSOR DOWN - SCREEN PLUS"          <E89>
300 CHAR ,6,24,"CURSOR UP{3SPACE}- SCREEN MINUS{H
    OME}"                                                  <0JH>
310 POKE BN,BI-256*INT(BI/256) : REM LOW-BYTE BIL
    DNUMMER BEKANNT GEBEN                                  <EKV>
320 POKE BN+1,INT(BI/256) : REM HIGH-BYTE BILDNUM
    MER                                                    <RSN>
330 SYS BA : REM BILDSCHIRM AUFBAUEN                       <E9C>
340 GET KEY A$ : REM AUF TASTENDRUCK WARTEN                <B47>
350 IF A$="{DOWN}" AND BI<14 THEN BI=BI+1: GOTO 3
    10: REM BILDNUMMER ERHOEHEN                            <NKH>
360 IF A$="{UP}" AND BI>0 THEN BI=BI-1: GOTO 31 :
     REM BILDNUMMER ERNIEDRIGEN                            <I15>
370 IF A$<>" " THEN 34 : REM AUSGANG BEI SPACE, S
    ONST WEITER                                            <0K9>
380 SYS IQ : REM ALTEN IRQ WIEDER EINSCHALTEN              <VFA>
390 SCNCLR : REM UND BILDSCHIRM LOESCHEN                   <LBR>
```

Listing 5. Mit diesem Demo-Programm (»DEMO .MSG«) werden MSG-Bilder per Basic verwaltet. Bitte mit dem Checksummer 128 eingeben.

```
Name : bildaufbau           0b00 0cff
-------------------------------------
0b00 : 20 81 0b ad 98 0c 85 fd   55
0b08 : a9 00 8d 99 0c ae 99 0c   fd
0b10 : bd 5a 0c 8d 98 0c 8a 0a   d7
0b18 : aa bd 9b 0c 85 fb 85 8e   75
0b20 : bd 9c 0c 85 fc 18 69 d4   bf
0b28 : 85 8f 20 27 0c ee 99 0c   19
0b30 : ad 99 0c c9 32 d0 d6 ad   46
0b38 : 8c 0c 48 29 0f 8d 93 0c   c5
0b40 : 68 6a 6a 6a 6a 29 0f 8d   0c
0b48 : 92 0c ad 8d 0c 29 01 d0   ad
0b50 : 03 a9 18 2c a9 1a 8d 94   7e
0b58 : 0c ad 8d 0c 29 02 d0 03   0c
0b60 : a9 c8 2c a9 d8 8d 95 0c   16
0b68 : ad 8e 0c 48 29 0f 8d 91   cd
0b70 : 0c 68 6a 6a 6a 6a 29 0f   55
0b78 : 8d 90 0c a5 fd 8d 98 0c   cc
0b80 : 60 8d 02 ff 20 ba 0b a0   6d
0b88 : 34 b1 fd 99 5a 0c 88 10   90
0b90 : f8 4c 52 0c a9 04 18 65   aa
0b98 : 87 85 87 90 02 e6 88 a9   a3
0ba0 : 04 18 65 89 85 89 90 02   26
0ba8 : e6 8a 18 a5 fb 69 28 85   45
0bb0 : fb 85 8e 90 04 e6 fc e6   5d
0bb8 : 8f 60 ad 96 0c 85 83 ad   0c
0bc0 : 97 0c 85 84 a9 35 20 fc   0e
0bc8 : 0b 18 a5 83 69 00 85 fd   62
0bd0 : a5 84 69 30 85 fe 60 85   f5
0bd8 : 83 a9 00 85 84 a9 20 20   37
0be0 : fc 0b 18 a5 83 69 00 85   ab
0be8 : 87 a5 84 69 10 85 88 a5   2b
0bf0 : 87 69 10 85 89 a5 88 69   9b
0bf8 : 00 85 8a 60 85 85 a2 00   79
0c00 : 86 86 86 8b 86 8c a0 10   4c
0c08 : d0 0d 18 a5 8b 65 85 85   1f
0c10 : 8b a5 8c 65 86 85 8c 46   91
0c18 : 8c 66 8b 66 84 66 83 88   22
0c20 : 30 04 90 f3 b0 e4 60 20   e9
0c28 : 4a 0c ad 98 0c 20 d7 0b   2e
0c30 : a2 03 a0 00 b1 87 91 fb   12
0c38 : b1 89 91 8e c8 c0 04 d0   28
0c40 : f3 20 94 0b ca 10 eb 4c   3f
0c48 : 52 0c 48 a9 7e 8d 00 ff   3c
0c50 : 68 60 48 a9 0e 8d 00 ff   7d
0c58 : 68 60 00 00 00 00 00 00   f1
0c60 : 00 00 00 00 00 00 00 00   61
0c68 : 00 00 00 00 00 00 00 00   69
0c70 : 00 00 00 00 00 00 00 00   71
0c78 : 00 00 00 00 00 00 00 00   79
0c80 : 00 00 00 00 00 00 00 00   81
0c88 : 00 00 00 00 00 00 00 07   97
0c90 : 02 06 00 00 18 c8 00 00   5d
0c98 : 00 00 d2 00 04 04 04 08   ce
0ca0 : 04 0c 04 10 04 14 04 18   cf
0ca8 : 04 1c 04 20 04 24 04 a0   72
0cb0 : 04 a4 04 a8 04 ac 04 b0   34
0cb8 : 04 b4 04 b8 04 bc 04 c0   e6
0cc0 : 04 c4 04 40 05 44 05 48   47
0cc8 : 05 4c 05 50 05 54 05 58   f6
0cd0 : 05 5c 05 60 05 64 05 e0   9a
0cd8 : 05 e4 05 e8 05 ec 05 f0   5b
0ce0 : 05 f4 05 f8 05 fc 05 00   0c
0ce8 : 06 04 06 80 06 84 06 88   30
0cf0 : 06 8c 06 90 06 94 06 98   1e
0cf8 : 06 9c 06 a0 06 a4 06 0b   96
```

Listing 6. Die Routine für den Bildaufbau in eigenen Programmen. Bitte mit dem MSE eingeben.

```
Name : irq                  1300 1393
-------------------------------------
1300 : 78 a9 24 8d 14 03 a9 13   2e
1308 : 8d 15 03 a9 01 8d 12 d0   7c
1310 : ad 11 d0 29 7f 8d 11 d0   e9
1318 : a9 81 8d 1a d0 a9 05 8d   b2
1320 : 06 d5 58 60 d8 20 31 13   ac
1328 : 20 87 fc 20 e7 c6 4c 33   9b
1330 : ff ad 19 d0 8d 19 d0 29   9d
1338 : 01 d0 01 60 ad 12 d0 cd   38
1340 : 9a 0c b0 2a ad 93 0c 8d   14
1348 : 20 d0 ad 92 0c 8d 21 d0   e1
1350 : ad 95 0c 8d 16 d0 ad 90   3c
1358 : 0c 8d 22 d0 ad 91 0c 8d   80
1360 : 23 d0 ad 94 0c 8d 18 d0   18
1368 : ad 9a 0c 4c 82 13 a9 15   81
1370 : 8d 18 d0 a9 00 8d 20 d0   01
1378 : 8d 21 d0 a9 c8 8d 16 d0   f2
1380 : a9 01 8d 12 d0 60 78 a9   95
1388 : 65 8d 14 03 a9 fa 8d 15   ec
1390 : 03 58 60 0b 38 e9 27 aa   fe
```

Listing 7. Die IRQ-Routine zum Einbau in eigene Programme. Bitte mit dem MSE eingeben.

Neue Zeichensätze für den 80-Zeichen-Modus

»Charedit« ermöglicht Ihnen das komfortable Editieren der Zeichensätze im 80-Zeichen-Modus. Eine grafische Benutzeroberfläche hilft bei der Bedienung. Zusätzlich bekommen Sie noch zwei neue Zeichensätze mitgeliefert.

Obwohl der DIN-Zeichensatz des C128 für den deutschen Anwender äußerst hilfreich ist, liefert er auf vielen Monitoren im 80-Zeichen-Modus ein eher mageres Bild. Unser Programm »Charedit« beseitigt dieses Manko, indem es gut lesbare Zeichensätze liefert, die vom Benutzer zudem komfortabel verändert und somit angepaßt werden können. Dies wird durch die freundliche Benutzerführung mit Windows und Pull-Down-Menüs sehr einfach gemacht. Des weiteren hat Charedit einige nützliche Routinen zu bieten, die auch einzeln verwendbar sind und auf diese Weise dem Hobbyprogrammierer ein individuelles Gestalten des Zeichensatzes ermöglichen. Zahlreiche Funktionen wie kopieren, spiegeln und rotieren von Zeichen sind dabei eine große Hilfe. Außerdem können gleichzeitig zwei Zeichensätze im Speicher sein, zwischen denen bei Bedarf hin- und hergeschaltet werden kann. Die Zeichensätze können als Dateien verwaltet und in eigene Programme eingebunden werden. Des weiteren bringt Charedit seine Ergebnisse schwarz auf weiß zu Papier.

Komfortable Pull-Down-Menüs

Das Programm selbst ist in Basic geschrieben. Bitte geben Sie Listing 1 mit dem Checksummer 128 ein. Beachten Sie dazu die Eingabehinweise auf Seite 159. Vor dem Starten mit RUN muß auf dem DIN-Zeichensatz umgeschaltet werden. Listing 2 bis 7 sind Maschinenroutinen, die mit dem MSE im C64-Modus eingegeben und vom Hauptprogramm automatisch nachgeladen werden. Dasselbe gilt für Listing 8 und 9 (die neuen Zeichensätze).

Wir wollen uns nun dem Editor zuwenden. Er arbeitet mit einer Benutzeroberfläche, die jener bekannter 16-Bit-Computer wie Atari ST oder Amiga ähnelt (Bild 1). Mittels Joystick erreicht man die verschiedenen Pull-Down-Menüs, aus denen per Knopfdruck der gewünschte Punkt ausgewählt wird. Zur direkten Eingabe von Daten werden Windows benützt, dadurch bleibt der Bildschirmaufbau übersichtlich.

Benutzeroberfläche wie beim Amiga

Auch bei der direkten Manipulation von Zeichen kann der Joystick hilfreich eingesetzt werden.

Nach dem Start mit »RUN« erscheint der »Mauszeiger« – ein weißer Pfeil – auf dem Bildschirm. Dieser wird mit einem Joystick in Port 2 gesteuert, kann aber auch mit einer (joysticksimulierenden) Maus bewegt werden. Sobald Sie mit diesem in ein Auswahlfeld gelangen und den Knopf am Joystick drücken, führt Charedit die gewählte Funktion aus. Zu Beginn kommen Sie durch Anklicken des »START«-Feldes in das Editorfeld. Mit »ENDE« können Sie Charedit verlassen.

Im Editor selbst werden die einzelnen Funktionen aus Pull-Down-Menüs aufgerufen. Durch Anklicken des entsprechenden Punktes in der Kopfleiste am Bildschirm erreichen Sie das dazugehörige Menü. In diesem erfolgt die Auswahl der einzelnen Funktionen analog. Wollen Sie ein Menü verlassen, weil Sie es versehentlich angewählt haben, müssen sie den Zeiger außerhalb des Menüs positionieren und den Feuerknopf am Joystick betätigen.

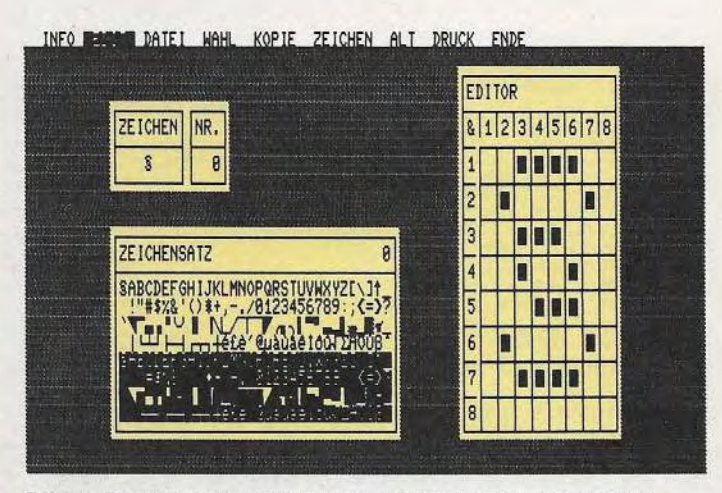


Bild 1. Die Benutzeroberfläche verleiht »Charedit« einen großen Bedienkomfort beim Editieren

Auf die vielseitigen Möglichkeiten die Ihnen Charedit bietet, soll nun im einzelnen eingegangen werden.

INFO

Gibt Auskunft über die aktuelle Programmversion und den Autor.

EDIT

In diesem Modus kann das angezeigte Zeichen punktweise verändert werden. Dazu positioniern Sie den Zeiger auf eines der Matrixfelder im Editor und klicken dieses an. Befand sich in dem Feld kein Punkt, wird dort einer gesetzt. War in diesem Feld bereits ein Punkt, so wird dieser gelöscht.

Diese Operation darf sich auch auf Gruppen von Punkten beziehen. Dazu dient die das Matrixfeld umgebende Zeilen- und Spaltennumerierung. Durch Anklicken einer dieser Zahlen wird die betreffende Spalte beziehungsweise Zeile invertiert, das heißt, gesetzte Punkte werden gelöscht und umgekehrt. Durch nochmaliges Anwählen dieser Funktion kann der ursprüngliche Zustand wiederhergestellt werden.

DATEI

Über diesen Menüpunkt werden die Diskettenfunktionen ausgeführt. Hier wird Ihr Zeichensatz gespeichert, wieder in den Speicher geladen und auf Wunsch gelöscht. Des weiteren erhalten Sie hier eine Liste aller auf der Diskette befindlichen Zeichensätze.

Die einzelnen Punkte wählen Sie durch Anklicken an. Versehentlich angewählte Funktionen werden durch Knopfdruck außerhalb des Menüfeldes wieder verlassen.

KATALOG

Durchsucht die im Laufwerk befindliche Diskette nach Zeichensatzdateien. Ein Window in Bildschirmmitte zeigt die Dateien an. Durch Anklicken der Felder »HOCH« und »RUNTER« blättern Sie die Liste durch. Wollen Sie den angezeigten Namen für eine im folgenden beschriebene Operation übernehmen, so klicken sie das »OK«-Feld an.

LADEN

Bei »LADEN«, »SPEICHERN« und »LÖSCHEN« erfolgt die Bedienung zweckmäßigerweise über die Tastatur. Um eine Funktion auszuführen, drücken Sie <RETURN>, um abzubrechen <ESC>.

Wenn Sie einen Zeichensatz laden wird der aktuelle im Speicher befindliche gelöscht. Der Name wird entweder über die Tastatur eingeben oder durch Druck auf <RETURN> direkt aus dem Window übernommen. Bei der Eingabe des Namens treten einige Besonderheiten auf: <DEL> löscht die komplette Zeile, es kann also nur durch die Cursortasten ein einzelnes Zeichen entfernt werden.

a) globale Variablen:	
a$	= allgemeine Stringvariable
I	= allgemeiner Zähler
KA$	= String für Directory (Katalog)
BL$	= String aus Leerzeichen
BG$	= String aus CHR$(221); Hintergrund
WO$	= Window-Off-String
DN$	= Dateiname
VD	= Basisadresse des VDC
PO	= Position des Pfeils
Adressen der Subroutinen:	
VP	= Startadresse von VDC-POKE
MC	= Startadresse von MODICHAR
TS	= Startadresse von TRANSSET
SS	= Startadresse von SWAPSCRN
AM	= Startadresse von CHARMOVE
SA	= Startadresse von SCNTOASC
X	= Zeiger auf das Programmende
AZ	= Aktuelles Zeichen
Z	= Hilfszeiger für AZ
AT	= Attributwert des Zeichens
C	= Zeichencode für SCNTOASC
C$	= ASCII-Zeichen von SCNTOASC
M	= Zeiger für den Modus
b) lokale Variablen der einzelnen Menüpunkte:	
INFO:	keine neuen Variablen
EDIT:	
L	= Zeiger für Zeile im Matrixfeld
C	= Zeiger für Spalte im Matrixfeld
ZF	= Array mit Matrix des Zeichens
DATEI:	
NS	= Nächster Sektor
TN	= Tracknummer
SK	= Sektornummer
Z	= Zählvariable
LZ	= Verringerte Zählvariable
N$	= Hilfsstring für Dateiname
L	= Länge der Eingabe
S	= Spaltenzeiger für Eingaben
AD	= Startadresse des Zeichensatzes in der Bitmap
AS	= Aktuelle Zeichensatznummer
FF	= Zeiger für Diskettenfehler
WAHL:	
Z	= Position des Zeichens im Satz
H	= Hilfsvariable für Z
A1	= Attributwert des neuen Zeichens
A2	= Attributwert des alten Zeichens
KOPIE:	keine neuen Variablen
ZEICHEN:	keine neuen Variablen
ALT:	keine neuen Variablen
DRUCK:	
O	= Offset für Zeichensatz
SP	= Zeiger für Spaces (Tabellierung)
J	= ASCII-Code des Zeichens
P	= Potenzierter Wert
ENDE:	keine neuen Variablen

Tabelle 1. Die Variablen von Charedit und deren Bedeutung

SPEICHERN

Schreibt den Zeichensatz im Speicher auf Diskette. Die Auswahl des Dateinamens erfolgt analog zu Laden.

LÖSCHEN

Entfernt bestimmte Zeichensätze von der Diskette. Diese Operation muß dem etwaigen Überschreiben von Dateien vorangehen, da der Speichervorgang bereits existierende Dateien nicht überschreibt. So sind Sie vor folgenschweren Fehlbedienungen geschützt.

Kommt es bei einer »DATEI«-Funktion zu Fehlern seitens des Floppylaufwerks, werden diese in einem Fenster angezeigt. Um die Arbeit mit Charedit fortführen zu können, müssen Sie in diesem Fall das Feld »BEHEBEN & BESTÄTIGEN« anklicken.

WAHL

Diese Funktion ermöglicht die Wahl des Zeichens aus dem aktuellen Zeichensatz, das bei »EDIT« verändert werden soll. Außerdem wird hier zwischen den beiden im Speicher befindlichen Zeichensätzen selektiert. Alle Funktionen beziehen sich stets auf den angezeigten Zeichensatz.

ZEICHEN WÄHLEN

Hier erscheint ein weiteres Menü. mit dessen Hilfe Sie das helle (also das aktuelle zu bearbeitende) Zeichen im aktuellen Zeichensatz bewegen. Es wird per Druck auf das »OK«-Feld in das Matrixfeld übernommen.

ZEICHENSATZ 0

Wählt Satz 0 als aktuellen Zeichensatz und zeigt diesen an.

ZEICHENSATZ 1

Macht das gleiche mit Zeichensatz 1.

KOPIE

Diese Funktion kopiert das Bitmuster des aktuellen Zeichens in ein beliebiges anderes Zeichen. Dadurch ersparen Sie sich das wiederholte Erstellen mehrmals vorkommender Grundelemente, die sich nur wenig voneinander unterscheiden.

Drehen, rotieren, invertieren – alles kein Problem

In einem Untermemü legen Sie fest, wohin Sie das weiß dargestellte Zeichen kopieren wollen. Durch Anklicken des Feldes »KOPIEREN« wird die Operation ausgeführt. Um das Menü wieder zu verlassen, klicken Sie das »OK«-Feld an.

ZEICHEN

Dieser Menüpunkt erhöht die Flexibilität von Charedit enorm. Hier kann das aktulle Zeichen auf verschiedenste Weise modifiziert werden.

LÖSCHEN

Löscht das aktuelle Zeichen.

INVERS

Kehrt das aktuelle Bitmuster um, das heißt, gesetzte Punkte werden gelöscht und umgekehrt.

SENKR.SPIEGELN

Spiegelt das Zeichen um seine senkrechte Mittelachse.

WAAGR.SPIEGELN

Spiegelt das Zeichen um seine waagrechte Mittelachse.

ROTIEREN (HOCH, LINKS, RECHTS, RUNTER)

Schiebt das Zeichen in die gewünschte Richtung. Dabei werden die herausfallenden Zeilen und Spalten am anderen Ende der Matrix wieder angefügt.
DREHEN (LINKS, RECHTS)
Dreht das Zeichen in die entsprechende Richtung um 90 Grad.

ALT
Versehentlich vorgenommene Änderungen an Zeichen werden hier rückgängig gemacht. Dies kann aber nur bis zu einem bestimmten Zeitpunkt erfolgen. Haben sie zwischenzeitlich den Zeichensatz abgespeichert oder auch nur gewechselt, so sind alle bis dahin erfolgten Manipulationen nicht mehr rückgängig zu machen.
ALTES ZEICHEN
Das aktuelle Zeichen wird auf seinen ursprünglichen Stand gebracht (soweit dies eben noch möglich ist).
ALTER Z-SATZ
Bringt den kompletten angezeigten Zeichensatz in seine ursprüngliche Form.

DRUCK
Bringt die Zeichensätze zu Papier. Die abgedruckte Version von Charedit arbeitet nur mit dem MPS 802 zusammen. Ein Anpassung ist allerdings leicht möglich, wir werden darauf noch zurückkommen.
ALTES=NEUES ZEICHEN
Druckt eine Gegenüberstellung der alten und neuen Zeichen.
HARDCOPY Z-SATZ
Gibt den Zeichensatz bildschirmgetreu auf den Drucker aus.

ENDE
Dient zum Verlassen von Charedit. Sicherheitshalber muß dies durch Anklicken des »OK«-Feldes bestätigt werden, da alle Daten beim Verlassen verlorengehen.

Zeichensätze als Beigabe

Mit Charedit lassen sich Zeichensätze aller Art eigenständig gestalten. Doch warum sollte man das Rad noch einmal erfinden? Wir haben Ihnen deshalb zwei Zeichensätze beigelegt, welche die Vielfältigkeit von Charedit demonstrieren. Listing 8 verleiht dem mageren DIN-Zeichensatz leserliche Buchstaben, und Listing 9 läßt den Bildschirm durch umrandete Zeichen lebendig erscheinen (siehe Bild 2). Auf der Programmservice-Diskette finden Sie noch weitere Zeichensätze. Mit Hilfe von speziellen Zeichensätzen lassen sich mit Charedit auch schöne Grafiken erstellen (Bild 3). Wir wünschen Ihnen viel Spaß bei der Arbeit mit Charedit!

Der Aufbau des Programms

Charedit wurde so programmiert, daß der erfahrene Anwender leicht Änderungen vornehmen kann. Betrachten Sie zu diesem Zweck die Verwendung der Variablen (Tabelle 1). Im Folgenden soll nun die Ansteuerung des Druckers erläutert werden, um die Anpassung der Ausgabefunktion an den eigenen Drucker zu ermöglichen.

Drucker einfach angepaßt

Die Druckroutine befindet sich am Ende des Programms aus Listing 1, dem Basic-Teil. Der Druck der gegenübergestellten Zeichen läuft in den Zeilen 12640 bis 12980 ab, die Hardcopy des Zeichensatzes erfolgt in den Zeilen 12990



Bild 2. Der Outline-Zeichensatz (Listing 9) als Ausdruck



Bild 3. Mit Hilfe von speziellen Zeichensätzen lassen sich mit »Charedit« auch solche Grafiken erstellen

bis 13300. Hier einige Erläuterungen, die ihnen beim Anpassen helfen sollen:
Zeile 12670: Öffnet den Druckkanal sowie den Kanal für die Sonderzeichen.
Zeile 12680: O = Offset für untere (Zeichen 0 bis 127) oder obere (Zeichen 128 bis 255) Hälfte des Zeichensatzes. SP = Zwischenraum der Zeichen (wird verändert, um Tabelle zu erzeugen).
Zeile 12730: Umrechnung Bildschirmcode in ASCII-Code.
Zeile 12740: Bitmuster des Zeichens holen.
Zeile 12890: Umgeformten Zeichenstring an Drucker senden; führt zum Ausdruck des Zeichens.
Zeile 12960: Druckerkanäle schließen.
Die übrigen Funktionen des Druckertreibers werden beim Durchsehen des Listings klar.

Routinen zum Übernehmen
Da Charedit den VDC-Zeichensatz manipuliert, verfügt das Programm über Routinen in Maschinensprache, die auch für sich betrachtet sehr interessant sind. Zur Übernahme von definierten Zeichensätzen in eigene Programme sind sie sogar obligatorisch. Wir werden die Listings 2 bis 7 im folgenden erläutern:

VDC-POKE - Listing 2:
Ändert den Inhalt einer Adresse im VDC-RAM, das 16 KByte umfaßt.
Programmbereich: $0C90 bis $0CBD
Syntax: SYS DEC("0C90"),L,H,W
Parameter:
L = Low-Byte
Low-Byte der Adresse
H = High-Byte
High-Byte der Adresse
W = Wert
Gewünschter Wert, mit dem die Adresse beschrieben werden soll.

MODICHAR - Listing 3:
Mit Modichar wird das Bitmuster eines bestimmten Zeichens im VDC ausgelesen oder nach Veränderung in den VDC geschrieben. Das Bitmuster muß acht Datenbytes umfassen und liegt im Bereich von $0C80 bis $0C87.
Programmbereich: $0C00 bis $0C88
Syntax: SYS DEC ("0C00"),Z,B,M
Parameter:



Z = Zeichensatznummer
Bei Z=1: Standardzeichensatz
Bei Z=2: Zweiter Zeichensatz (über <SHIFT CBM> erreichbar)
B = Bildschirmcode
Nummer des ausgesuchten Zeichens im Bildschirmcode
M = Modus
Bei M=0: holt das Zeichen aus dem VDC
Bei M=1: schreibt das Zeichen in den VDC

TRANSSET - Listing 4:

Mit Transset wird ein VDC-Zeichensatz ausgelesen oder überschrieben. Beim Auslesen holt das Programm das Muster aus der Bitmap. Beim Überschreiben legt es dieses dort hin. Hier kann der Zeichensatz problemlos auf Diskette geschrieben, beziehungsweise von dort geholt werden. Da der VDC über zwei Zeichensätze verfügt, werden diese an zwei verschiedenen Stellen in der Bitmap abgelegt (bei \$2000 beziehungsweise \$3000). Transset überträgt die Zeichen optimierend. Dazu werden die acht überflüssigen Nullbytes, die der VDC jedem Zeichen anhängt, beim Lesen gelöscht und beim Schreiben wieder angehängt.
Programmbereich: \$0B00 bis \$0BA6
Syntax: SYS DEC ("0B00"),Z,B,M
Parameter:
Z = Zeichensatznummer
Bei Z=1: Standardzeichensatz
Bei Z=2: Zweiter Zeichensatz
B = Bitmapposition
Bestimmt die Position des Zeichensatzes
Bei B=0: Zeichensatz ab \$2000
Bei B=1: Zeichensatz ab \$3000
M = Modus
Bei M=0: Schreibt Zeichensatz aus der Bitmap in den VDC.
Bei M=1: Liest VDC-Zeichensatz und schreibt ihn in die Bitmap.

Transset eignet sich besonders zum Laden von Zeichensaätzen, die mit Charedit erstellt wurden.
Beispiel:

```
10 BLOAD "TRANSSET.OBJ"
20 BLOAD "[Name des Zeichensatzes]*", P 8192
30 SYS DEC ("0B00"),[Zeichensatznummer],0,0
```

SWAPSCRN - Listing 5:

Diese Routine ermöglicht ein Zwischenspeichern des Bildschirminhalts im 80-Zeichen-Modus. Dazu wird der Inhalt des 80-Zeichen-Bildschirms mitsamt seiner Attribute in den freien Bereich des VDC-RAMs ab \$1000 verschoben. Oder der sichtbare Bildschirm wird durch den Inhalt dieses RAMs ersetzt. So können sie den sichtbaren Bildschirm nach vorher erfolgter Sicherung beliebig »verunstalten«, denn der alte ursprüngliche Zustand ist problemlos wiederherzustellen.
Programmbereich: \$0D00 bis \$0D90
Syntax: SYS DEC("0D00"),M
Parameter:
M = Modus
Bei M=0: Holt den gespeicherten Bildschirminhalt wieder auf den sichtbaren Bildschirm
Bei M=1: Speichert den sichtbaren Bildschirm

CHARMOVE - Listing 6:

Bewegt ein beliebiges Zeichen des VDC-Zeichensatzes über den 80-Zeichen-Bildschirm. Das Zeichen wird mit dem Joystick in Port 2 gesteuert.
Programmbereich: \$1300 bis \$14C5
Syntax: SYS DEC ("1300")
Parameter:
Um eine Übergabe der zahlreichen Parameter schnell

ADRESSE	BEDEUTUNG
\$14BF	Position des Zeichens: Zeile (0 bis 24)
\$14C0	Spalte (0 bis79)
\$14C1	Bildschirmcode des Zeichens (0 bis 255)
\$14C2	Attribute des Zeichens Bit 7: 0; Zeichensatz 0 1; Zeichensatz 1 Bit 6: 0; Zeichen normal 1; Zeichen revers Bit 5: 0; Unterstreichen aus 1; Unterstreichen ein Bit 4: 0; Blinken aus 1; Blinken ein Bit 3: Farbe des Zeichens (0 bis 15) bis Bit 0
\$14C3	Tempo der Zeichenbewegung: vertikal (0 bis 255)
\$14C4	horizontal (0 bis 255)

und problemlos zu ermöglichen, befinden sich diese in folgenden Speicherzellen:

Nach Drücken des Joystickknopfes kann die aktuelle Position des Zeichens gelesen werden.

SCNTOASC - Listing 7:

Wandelt Bildschirmcode in ASCII-Code. Der Bildschirmcode wird als Parameter übergeben, der ASCII-Code kann mit »PEEK (DEC("FA"))« oder »RREG« gelesen werden.
Programmbereich: \$0CC0 bis \$0CE4
Syntax: SYS DEC("0CC0"),B
Parameter:
B = Bildschirmcode des gewünschten Zeichens

(Gerhard Zinner/Axel Pretzsch)

Achtung!

Bitte beachten Sie bei der Eingabe der Zeichensätze (Listing 8 und 9) mit dem MSE (im C64-Modus!), daß der Name des jeweiligen Zeichensatzes vollständig eingegeben wird, also auch alle Spaces und das Grafikzeichen <CBM> an der 16ten Position. Bei Listing 8 muß also »fett (g/k) <5mal SPACE> <CBM B>«, bei Listing 9 »outline (g/k) <2mal SPACE> <CBM B>« eingegeben werden.

Dies ist deshalb notwendig, da Charedit die Zeichensätze als solche nur erkennt, wenn am Ende des Namens das über <CBM B> zu erreichende Grafikzeichen steht.

Wenn der Zeichensatz hingegen von Charedit aus geladen wird, reicht es, den Namen des jeweiligen Zeichensatzes anzugeben.

Bei der Eingabe von Listing 1 mit dem Checksummer darf der DIN-Zeichensatz nicht aktiviert sein. Eine Code-Umwandlung (wie im Kopf von Listing 1 angegeben) braucht dann nicht beachtet zu werden.

```
1000 REM  ******************************        <34C>
1010 REM                                        <3CD>
1020 REM         CHAREDIT  V1.0                 <3KE>
1030 REM         --------------                 <3SF>
1040 REM  EIN ZEICHENSATZEDITOR FUER DEN        <248>
1050 REM  80-ZEICHEN-MODUS DES C128 UND         <2C9>
1060 REM  DRUCKER VC 1526 / MPS 802.            <2KA>
1070 REM                                        <2SB>
1080 REM  01.87                 G. ZINNER       <144>
1090 REM                                        <1C5>
1100 REM  DAS PRG WURDE IM DIN-MODUS ER-        <3KE>
1110 REM  STELLT. DARUM SIND BEI FOLGEN-        <3SF>
1120 REM  DEN GRAFIKZEICHEN DIESE TASTEN        <34C>
1130 REM  ZU BETAETIGENCHEN :                   <1G7>
1140 REM                                        <2KA>
1150 REM  "┬" = CBM + A / "┐" = CBM + S         <2SB>
1160 REM  "┬" = CBM + Y / "┘" = CBM + X         <248>
1170 REM  "┐" = CBM + B / "┼" = CBM + +         <2C9>
1180 REM  "┘" = CBM + Q / "┤" = CBM + W         <1K6>
1190 REM  "═" = CBM + # / "┘" = CBM + ?         <1S7>
1200 REM                                        <248>
1210 REM  ******************************        <2C9>
1220 :                                          <209>
1230 FAST                                       <MIJ>
1240 :                                          <105>
1250 : REM - SCHALTET AUF 80 ZEICHEN            <1G7>
```

Listing 1. Das Basic-Programm »CHAREDIT«. Es lädt die Unterprogramme aus Listing 2 bis 8 selbständig nach. Bitte mit dem Checksummer 128 eingeben.

```
1260 :                                                 <001>
1270 IF PEEK(DEC("D7"))=0 THEN PRINT CHR$(27)+"X"      <B7K>
1280 :                                                 <70T>
1290 : REM - PRUEFT AUF ASCII-Z-SATZ                   <7GV>
1300 :                                                 <209>
1310 IF PEEK(DEC("CC"))=128 THEN BEGIN                 <U21>
1320 : SCNCLR                                          <TLJ>
1330 : A$="=": FOR I=0 TO 79: A$=A$+"#": NEXT I        <F96>
1340 : PRINT : PRINT : PRINT                           <QLF>
1350 : PRINT USING A$;"SCHALTEN SIE AUF DEN DIN-Z
     EICHENSATZ UM"                                    <I8T>
1360 : PRINT USING A$;"UND STARTEN SIE DAS PROGRA
     MM"                                               <CV8>
1370 : PRINT USING A$;"MIT 'RUN' - <RETURN> NEU !
     "                                                 <7D9>
1380 : END                                             <9HE>
1390 BEND                                              <LUU>
1400 :                                                 <50L>
1410 : REM - 80-Z CHARACTERRAM INIT                    <5GN>
1420 :                                                 <40H>
1430 SYS DEC("CE0C")                                   <7SF>
1440 :                                                 <70T>
1450 : REM - GRAPHICBEREICH FREIMACHEN                 <7GV>
1460 :                                                 <60P>
1470 GRAPHIC 1,1: GRAPHIC 0: GRAPHIC 5                 <DPR>
1480 :                                                 <105>
1490 : REM - VARIABLEN VORBESETZEN                     <1G7>
1500 :                                                 <40H>
1510 DIM KA$(144)                                      <BN8>
1520 FOR I=0 TO 79                                     <U0N>
1530 : BL$=BL$+CHR$(32)                                <E9C>
1540 : BG$=BG$+CHR$(221)                               <22F>
1550 NEXT I                                            <LQ4>
1560 WO$=CHR$(19)+CHR$(19)                             <874>
1570 DN$=LEFT$(BL$,15)                                 <374>
1580 VD=DEC("D600")                                    <KC2>
1590 DEF FN PO(P)=PEEK(DEC("14BF"))*80+PEEK(DEC("
     14C0"))                                           <8M7>
1600 :                                                 <70T>
1610 : REM - ML-PROGS NACHLADEN                        <7GV>
1620 :                                                 <60P>
1630 BLOAD "CHARMOVE.OBJ"                              <6HD>
1640 BLOAD "MODICHAR.OBJ"                              <2I5>
1650 BLOAD "TRANSSET.OBJ"                              <7FD>
1660 BLOAD "SWAPSCRN.OBJ"                              <S5L>
1670 BLOAD "SCNTOASC.OBJ"                              <J6D>
1680 BLOAD "VDC-POKE.OBJ"                              <36L>
1690 :                                                 <3GF>
1700 : REM - ML-VARIABLEN VORBESETZEN                  <60P>
1710 :                                                 <6GR>
1720 AM=DEC("1300") : REM CHARMOVE                     <DBC>
1730 TS=DEC("0B00") : REM TRANSSET                     <GEV>
1740 MC=DEC("0C00") : REM MODICHAR                     <S2E>
1750 SS=DEC("0D00") : REM SWAPSCRN                     <GHS>
1760 SA=DEC("0CC0") : REM SCNTOASC                     <2AK>
1770 VP=DEC("0C90") : REM VDC-POKE                     <QS5>
1780 :                                                 <209>
1790 : REM - ZEICHENSAETZE VERSCHIEBEN                 <2GB>
1800 :                                                 <915>
1810 SYS TS,0,0,1                                      <OM0>
1820 SYS TS,1,1,1                                      <S6A>
1830 SYS TS,1,0,0                                      <UI8>
1840 :                                                 <B1D>
1850 : REM - AUS @ WIRD DER PFEIL                      <BHF>
1860 :                                                 <A19>
1870 SYS MC,0,0,1                                      <5K0>
1880 :                                                 <D1L>
1890 :                                                 <DHN>
1900 : REM ============================                <811>
1910 : REM  TITELBILD                                  <8H3>
1920 : REM ============================                <915>
1930 :                                                 <9H7>
1940 :                                                 <A19>
1950 COLOR 6,3: COLOR 5,1                              <505>
1960 PRINT CHR$(27)+"M"                                <VEE>
1970 SCNCLR                                            <4TJ>
1980 :                                                 <C1H>
1990 PRINT CHR$(2);BL$                                 <IEG>
2000 COLOR 5,13                                        <E45>
2010 FOR I=0 TO 23: PRINT BG$: NEXT I                  <USE>
2020 COLOR 5,1                                         <4KE>
2030 :                                                 <6GR>
2040 WINDOW 14,3,64,13,1                               <L08>
2050 : PRINT "£++++++++++++++++++++++++++++++++++
     ++++++++++++++N"                                  <EK3>
2060 : PRINT "K{49SPACE}K"                             <06V>
2070 : PRINT "K{2SPACE}_____ _{3SPACE}_ _____ ___
     __ _____ ____{2SPACE}___ _____{2SPACE}K"          <HCR>
2080 : PRINT "K{2SPACE}_{5SPACE}_{3SPACE}_ _{3SPA
     CE}_ _{3SPACE}_ _{5SPACE}_{3SPACE}_{2SPACE}_
     {4SPACE}_{4SPACE}K"                               <REE>
2090 : PRINT "K{2SPACE}_{5SPACE}_____ _____ _____
      _____ _{3SPACE}_{2SPACE}_{4SPACE}_{4SPACE}K
     "                                                 <NPE>
2100 : PRINT "K{2SPACE}_{5SPACE}_{3SPACE}_ _{3SPA
     CE}_ _{2SPACE}_{2SPACE}_{5SPACE}_{3SPACE}_{2
     SPACE}_{4SPACE}_{4SPACE}K"                        <CEC>
2110 : PRINT "K{2SPACE}_____ _{3SPACE}_ _{3SPACE}
     _ _{3SPACE}_ _____ ____{2SPACE}___{3SPACE}_{
     4SPACE}K"                                         <OGR>
2120 : PRINT "K{49SPACE}K"                             <C6V>
2130 : PRINT "K{3SPACE}E I N{5SPACE}Z E I C H E N
      S A T Z E D I T O R{3SPACE}K"                    <OO7>
2140 : PRINT "K{49SPACE}K"                             <06V>
2150 : PRINT "T++++++++++++++++++++++++++++++++++
     ++++++++++++++++G"                                <355>
2160 PRINT WO$                                         <7K3>
2170 :                                                 <5GN>
2180 WINDOW 14,16,47,22,1                              <DC4>
2190 : PRINT "£++++++++++++++++++++++++++++++++N"      <9C1>
2200 : PRINT "K SCHLIESSEN SIE AN PORT 2 EINEN K"      <NOG>
2210 : PRINT "K JOYSTICK ODER EINE MOUSE AN,{3SPA
     CE}K"                                             <A3L>
2220 : PRINT "K UND BEWEGEN SIE DEN PFEIL ( ){2SP
     ACE}K"                                            <6TD>
2230 : PRINT "K AUF EINES DER FELDER UND KLIK- K"      <96B>
2240 : PRINT "K KEN SIE DIESES AN.{13SPACE}K"          <1EQ>
2250 : PRINT "T++++++++++++++++++++++++++++++++G"      <5SR>
2260 PRINT WO$                                         <643>
2270 :                                                 <6GR>
2280 WINDOW 56,16,64,18,1                              <HES>
2290 : PRINT "£+++++++N"                               <7J1>
2300 : PRINT "K ENDE{2SPACE}K"                         <2KG>
2310 : PRINT "T+++++++G"                               <3JR>
2320 PRINT WO$                                         <7K3>
2330 :                                                 <5GN>
2340 WINDOW 56,20,64,22,1                              <BDK>
2350 : PRINT "£+++++++N"                               <0JT>
2360 : PRINT "K START K"                               <UB4>
2370 : PRINT "T+++++++G"                               <0JN>
2380 PRINT WO$                                         <543>
2390 :                                                 <0G3>
2400 : REM - SYSTEMSTART ODER -ENDE                    <30D>
2410 :                                                 <3GF>
2420 X=0                                               <GE5>
2430 DO                                                <JEF>
2440 : SYS AM: PO=FN PO(0)                             <MHK>
2450 : IF (PO>1416 AND PO<1425) OR (PO>1496 AND P
     O<1505) THEN BEGIN                                <988>
2460 :   SCNCLR : PRINT CHR$(27)+"L": END              <30C>
2470 : BEND                                            <JUU>
2480 : IF (PO>1736 AND PO<1745) OR (PO>1816 AND P
     O<1825) THEN X=1                                  <IUQ>
2490 LOOP WHILE X=0                                    <UHO>
2500 :                                                 <209>
2510 :                                                 <2GB>
2520 : REM  ============================               <30D>
2530 : REM  ARBEITSSCREEN AUFBAUEN                     <3GF>
2540 : REM  ============================               <001>
2550 :                                                 <0G3>
2560 :                                                 <105>
2570 PRINT CHR$(19);CHR$(2);                           <J1M>
2580 PRINT "{2SPACE}INFO{2SPACE}EDIT{2SPACE}DATEI
     {2SPACE}WAHL{2SPACE}KOPIE{2SPACE}ZEICHEN{2SP
     ACE}ALT{2SPACE}DRUCK{2SPACE}ENDE"                 <PE7>
2590 COLOR 5,13                                        <FK5>
2600 FOR I=0 TO 23: PRINT LEFT$(BG$,79): NEXT I        <DN5>
2610 COLOR 5,1                                         <7CE>
2620 :                                                 <001>
2630 WINDOW 10,4,23,8,1                                <JJ5>
2640 : PRINT "£+++++++N£+++N"                          <SOG>
2650 : PRINT "KZEICHENKKNR.K"                          <R50>
2660 : PRINT "G+++++++MG+++M"                          <BJC>
2670 : PRINT "K{7SPACE}KK{3SPACE}K"                    <P01>
2680 : PRINT "T+++++++GT+++G"                          <NJV>
2690 PRINT WO$                                         <7S3>
2700 :                                                 <001>
2710 WINDOW 10,11,43,22,1                              <JJC>
2720 : PRINT "£++++++++++++++++++++++++++++++++N"      <EC1>
2730 : PRINT "KZEICHENSATZ{21SPACE}K"                  <RA2>
2740 : PRINT "G++++++++++++++++++++++++++++++++M"      <DKM>
2750 : FOR I=0 TO 7                                    <RQ6>
2760 : PRINT "K";LEFT$(BL$,32);"K"                     <GC4>
2770 : NEXT I                                          <PM4>
2780 : PRINT "T++++++++++++++++++++++++++++++++G"      <2SR>
2790 PRINT WO$                                         <7C3>
2800 SYS VP,234,3,48+AS                                <VMC>
2810 :                                                 <FHV>
2820 WINDOW 11,14,42,21,1                              <DGI>
2830 : PRINT CHR$(14);                                 <5GG>
2840 : FOR I=0 TO 31: PRINT CHR$(I+64);: NEXT I:
     PRINT                                             <1D8>
2850 : PRINT CHR$(32);CHR$(33);CHR$(34);CHR$(20);
     CHR$(34);                                         <EM4>
2860 : FOR I=35 TO 63: PRINT CHR$(I);: NEXT I: PR
     INT                                               <SI6>
2870 : FOR I=64 TO 95: PRINT CHR$(I+32);: NEXT I:
      PRINT                                            <LQ3>
2880 : FOR I=96 TO 127: PRINT CHR$(I+64);: NEXT I
     : PRINT                                           <JQI>
2890 : PRINT CHR$(18);: FOR I=0 TO 31: PRINT CHR$
     (I+64);: NEXT I                                   <4II>
2900 : PRINT CHR$(18);CHR$(32);CHR$(33);CHR$(34);
     CHR$(20);CHR$(34);                                <51U>
2910 : FOR I=35 TO 63: PRINT CHR$(I);: NEXT I          <DKE>
2920 : PRINT CHR$(18);: FOR I=64 TO 95: PRINT CHR
     $(I+32);: NEXT I                                  <PT5>
2930 : PRINT CHR$(18);: FOR I=96 TO 127: PRINT CH
     R$(I+64);: NEXT I                                 <FJ7>
2940 : PRINT CHR$(142)                                 <1K0>
2950 PRINT WO$                                         <3C3>
2960 :                                                 <D1L>
2970 WINDOW 51,2,69,22,1                               <L0I>
2980 : PRINT "£+++++++++++++++++N"                     <GG3>
2990 : PRINT "KEDITOR{11SPACE}K"                       <9EI>
3000 : PRINT "G+£+£+£+£+£+£+£+£+M"                     <2TQ>
3010 : PRINT "K&K1K2K3K4K5K6K7K8K"                     <BJE>
3020 : PRINT "G+£+£+£+£+£+£+£+£+M"                     <6LR>
```

```
3030 : FOR I=1 TO 7                                    <MQ2>
3040 : PRINT "K";RIGHT$(STR$(I),1);"K K K K K K K
       K K"                                            <CP3>
3050 : PRINT "G+G+G+G+G+G+G+G+G+M"                     <BCU>
3060 : NEXT I                                          <904>
3070 : PRINT "KGK K{SHF.SPACE}K{SHF.SPACE}K{SHF.S
       PACE}K{SHF.SPACE}K{SHF.SPACE}K{SHF.SPACE}K{S
       HF.SPACE}K"                                     <M45>
3080 : PRINT "I+I+I+I+I+I+I+I+I+@"                     <L55>
3090 PRINT WO$                                         <5S3>
3100 :                                                 <40H>
3110 : REM - VORBESETZUNGEN                            <4GJ>
3120 :                                                 <50L>
3130 AZ=0: Z=AZ: AT=143: GOSUB 12210: GOSUB 11830      <6M8>
3140 CHAR ,7,0," EDIT ",1                              <J3D>
3150 X=0                                               <2CD>
3160 :                                                 <70T>
3170 :                                                 <7GV>
3180 : REM  ===============                            <001>
3190 : REM  GROSSE SCHLEIFE                            <0G3>
3200 : REM  ===============                            <70T>
3210 :                                                 <7GV>
3220 :                                                 <60P>
3230 DO                                                <RFF>
3240 :                                                 <50L>
3250 : REM - ANZEIGEN AKTUALISIEREN                    <5GN>
3260 :                                                 <40H>
3270 : C=AZ: GOSUB 11750                               <4AE>
3280 : WINDOW 14,7,14,7,1                              <6TN>
3290 :  PRINT CHR$(14);C$;CHR$(142)                   <FPA>
3300 : PRINT WO$                                       <0G2>
3310 : CHAR ,20,7: PRINT USING ">##";AZ                <6G6>
3320 :                                                 <70T>
3330 : REM - PFEILSTEUERROUT ANSPRINGEN                <7GV>
3340 :                                                 <40H>
3350 : SYS AM: PO=FN PO(0)                             <J12>
3360 :                                                 <50L>
3370 :                                                 <5GN>
3380 : REM =====================                       <209>
3390 : REM PFEIL IM EDITORFELD ?                       <2GB>
3400 : REM =====================                       <105>
3410 :                                                 <1G7>
3420 IF PO>80 THEN BEGIN                               <G1N>
3430 :                                                 <0G3>
3440 : REM  ============================               <30D>
3450 : REM  EDIT                                       <3GF>
3460 : REM  ============================               <209>
3470 :                                                 <2GB>
3480 : L=255                                           <NVV>
3490 :                                                 <5GN>
3500 : IF PO>453 AND PO<470 THEN PO=PO-454: L=-1       <NNU>
3510 : IF PO>533 AND PO<550 THEN PO=PO-534: L=-1       <RJE>
3520 : IF PO>611 AND PO<630 THEN PO=PO-612: L=0        <2CT>
3530 : IF PO>691 AND PO<710 THEN PO=PO-692: L=0        <88L>
3540 : IF PO>771 AND PO<790 THEN PO=PO-772: L=1        <UBT>
3550 : IF PO>851 AND PO<870 THEN PO=PO-852: L=1        <04L>
3560 : IF PO>931 AND PO<950 THEN PO=PO-932: L=2        <AAC>
3570 : IF PO>1011 AND PO<1030 THEN PO=PO-1012: L=
       2                                               <AUE>
3580 : IF PO>1091 AND PO<1110 THEN PO=PO-1092: L=
       3                                               <Q4E>
3590 : IF PO>1171 AND PO<1190 THEN PO=PO-1172: L=
       3                                               <25E>
3600 : IF PO>1251 AND PO<1270 THEN PO=PO-1252: L=
       4                                               <Q4D>
3610 : IF PO>1331 AND PO<1350 THEN PO=PO-1332: L=
       4                                               <A4D>
3620 : IF PO>1411 AND PO<1430 THEN PO=PO-1412: L=
       5                                               <I5D>
3630 : IF PO>1491 AND PO<1510 THEN PO=PO-1492: L=
       5                                               <27D>
3640 : IF PO>1571 AND PO<1590 THEN PO=PO-1572: L=
       6                                               <Q6C>
3650 : IF PO>1651 AND PO<1670 THEN PO=PO-1652: L=
       6                                               <27C>
3660 : IF PO>1731 AND PO<1750 THEN PO=PO-1732: L=
       7                                               <I7C>
3670 : IF PO>1811 AND PO<1830 THEN PO=PO-1812: L=
       7                                               <Q4D>
3680 :                                                 <70T>
3690 : REM - AENDERUNGEN AUSFUEHREN                    <7GV>
3700 :                                                 <209>
3710 : IF L<>255 THEN BEGIN                            <7KC>
3720 :                                                 <30D>
3730 : REM - SPALTE UNDIEREN                           <3GF>
3740 :                                                 <001>
3750 :  IF L=-1 THEN BEGIN                             <BHA>
3760 :   M=1: C=INT(PO/2)                              <MMD>
3770 :   FOR L=0 TO 7: GOSUB 3910: NEXT L              <7AH>
3780 :   PO=-1                                         <SQJ>
3790 :  BEND                                           <JIU>
3800 :                                                 <D1L>
3810 : REM - ZEILE UNDIEREN                            <DHN>
3820 :                                                 <C1H>
3830 :  IF PO>-1 AND PO<2 THEN BEGIN                   <6CF>
3840 :   M=1: FOR C=0 TO 7: GOSUB 3910: NEXT C         <901>
3850 :  BEND                                           <30U>
3860 :                                                 <E1P>
3870 : REM - PUNKTE SETZEN/LOESCHEN                    <EHR>
3880 :                                                 <915>
3890 :  IF PO>1 THEN BEGIN                             <FPP>
3900 :   M=0: C=INT(PO/2)-1                            <5NB>
3910 :   ZF(L,C)=XOR(ZF(L,C),1)                        <NNF>
3920 :   CHAR ,54+(C*2),7+(L*2)," ",ZF(L,C)            <20N>
3930 :   D=PEEK(DEC("0C80")+L)                         <PGP>
3940 :   IF ZF(L,C)=0 THEN D=D AND (255-2↑(7-C)):
      ELSE D=D OR 2↑(7-C)                              <JUA>
3950 :   POKE DEC("0C80")+L,D                          <82V>
3960 :   SYS MC,1,AZ,1                                 <QT6>
3970 :   IF M=1 THEN RETURN                            <HSN>
3980 :  BEND                                           <RFU>
3990 :                                                 <8H3>
4000 : REM - ENDE AUSFUEHREN                           <B1D>
4010 :                                                 <BHF>
4020 : BEND                                            <RBU>
4030 : PO=80                                           <846>
4040 :                                                 <915>
4050 : REM - ENDE PO>80                                <9H7>
4060 :                                                 <811>
4070 BEND                                              <5GU>
4080 :                                                 <F1T>
4090 :                                                 <FHV>
4100 : REM =====================                       <A19>
4110 : REM PFEIL IN MENUELEISTE ?                      <AHB>
4120 : REM =====================                       <B1D>
4130 :                                                 <BHF>
4140 IF PO<80 THEN BEGIN                               <M17>
4150 :                                                 <8H3>
4160 : REM - EDIT AUSSCHALTEN                          <915>
4170 :                                                 <9H7>
4180 : CHAR ,7,0,CHR$(2)+" EDIT "+CHR$(13)             <S60>
4190 :                                                 <EHR>
4200 : REM - BILDSCHIRM RETTEN                         <915>
4210 :                                                 <9H7>
4220 : SYS SS,1                                        <AHM>
4230 :                                                 <8H3>
4240 : REM  ============================               <B1D>
4250 : REM  INFO                                       <BHF>
4260 : REM  ============================               <A19>
4270 :                                                 <AHB>
4280 IF PO>1 AND PO<6 THEN BEGIN                       <HB2>
4290 :                                                 <DHN>
4300 : CHAR ,1,0," INFO ",1                            <KOT>
4310 :                                                 <8H3>
4320 : WINDOW 26,8,52,15,1                             <LD0>
4330 :  PRINT "£++++++++++++++++++++++++++N"           <UG6>
4340 :  PRINT "K NAME{4SPACE}: CHAREDIT{6SPACE}K"      <SL4>
4350 :  PRINT "K VERSION : V 1.0{9SPACE}K"             <KRH>
4360 :  PRINT "K DATUM{3SPACE}: 10.86{9SPACE}K"        <6HJ>
4370 :  PRINT "K AUTOR{3SPACE}: G. ZINNER{5SPACE}
       K"                                              <OON>
4380 :  PRINT "K{11SPACE}SCHMALE STR.3 K"              <KV7>
4390 :  PRINT "K{11SPACE}4300 ESSEN 11 K"              <IBD>
4400 :  PRINT "I++++++++++++++++++++++++++@"           <D30>
4410 : PRINT WO$                                       <902>
4420 :                                                 <E1P>
4430 : SYS AM                                          <UET>
4440 : SYS SS,0                                        <85M>
4450 :                                                 <DHN>
4460 BEND                                              <1KE>
4470 :                                                 <CHJ>
4480 : REM  ============================               <B1D>
4490 : REM  DATEI                                      <BHF>
4500 : REM  ============================               <E1P>
4510 :                                                 <EHR>
4520 IF PO>13 AND PO<19 THEN BEGIN                     <829>
4530 :                                                 <FHV>
4540 : M=0                                             <FHN>
4550 : CHAR ,13,0," DATEI ",1                          <45V>
4560 :                                                 <D1L>
4570 : WINDOW 13,1,25,9,1                              <JB7>
4580 :  PRINT "K{11SPACE}K"                           <J7G>
4590 :  PRINT "K KATALOG{3SPACE}K"                     <HU1>
4600 :  PRINT "K++++++++++++K"                         <KDJ>
4610 :  PRINT "K LADEN{5SPACE}K"                       <40H>
4620 :  PRINT "K++++++++++++K"                         <4FJ>
4630 :  PRINT "K SPEICHERN K"                          <E39>
4640 :  PRINT "K++++++++++++K"                         <K9J>
4650 :  PRINT "K LOESCHEN{3SPACE}K"                    <DUG>
4660 :  PRINT "I++++++++++++@"                         <93M>
4670 : PRINT WO$                                       <802>
4680 :                                                 <915>
4690 : SYS AM : PO=FN PO(0)                            <U0M>
4700 :                                                 <C1H>
4710 : REM - KATALOG                                   <CHJ>
4720 :                                                 <D1L>
4730 : IF (PO>173 AND PO<186) OR (PO>253 AND PO<2
       66) THEN BEGIN                                  <9LE>
4740 :                                                 <E1P>
4750 :  WINDOW 27,8,57,17,1                            <U9C>
4760 :   PRINT "£+++++++++++++++++++++++++++++N"       <AF0>
4770 :   PRINT "K KATALOG{21SPACE}K"                   <A69>
4780 :   PRINT "G+++++++++++++++++++++++++++++M"       <07E>
4790 :   PRINT "K{29SPACE}K"                           <0OD>
4800 :   PRINT "K{29SPACE}K"                           <24D>
4810 :   PRINT "K{29SPACE}K"                           <20D>
4820 :   PRINT "K£++++++++N£++++++++N{3SPACE}£+++
       +NK"                                            <VU1>
4830 :   PRINT "KK{2SPACE}HOCH{2SPACE}KK RUNTER K
       {3SPACE}K OK KK"                                <VES>
4840 :   PRINT "KI++++++++@I++++++++@{3SPACE}I+++
       +@K"                                            <AVC>
4850 :   PRINT "I+++++++++++++++++++++++++++++@"       <AVC>
4860 :  PRINT WO$                                      <6G2>
```

Listing 1. (Fortsetzung)

```
4870 :                                                  <0G3>
4880 : REM - DIRECTORY IN KA$() LADEN                   <70T>
4890 :                                                  <7GV>
4900 :  CHAR ,29,12,"{2SPACE}LESE DATEI NR. ...."       <KTT>
4910 :  OPEN 1,8,15: OPEN 8,8,8,"#0"                    <IPS>
4920 :  PRINT#1,"U1:";8;0;18;0                          <0KB>
4930 :  GOSUB 11960                                     <4UK>
4940 :                                                  <001>
4950 :  IF FF=0 THEN BEGIN                              <FLB>
4960 :   NS=1: Z=0                                      <NM0>
4970 :                                                  <1G7>
4980 :   DO                                             <0S7>
4990 :    TR=18: SK=NS                                  <TI1>
5000 :    PRINT#1,"U1:";8;0;TR;SK                       <LAG>
5010 :    GET #8,A$,A$: NS=ASC(A$)                      <ARE>
5020 :    DATA 2,34,66,98,130,162,194,226,-1            <059>
5030 :    RESTORE 5020                                  <D3V>
5040 :                                                  <B1D>
5050 :    DO                                            <MQV>
5060 :     READ D                                       <PI0>
5070 :                                                  <AHB>
5080 :     IF D>0 THEN BEGIN                            <PBE>
5090 :      PRINT#1,"B-P:",8,D                          <H44>
5100 :      GET #8,A$                                   <RMI>
5110 :                                                  <8H3>
5120 :      IF ASC(A$)<>0 AND ASC(A$)<>128 THEN B
     EGIN                                               <650>
5130 :       GET #8,A$,A$                               <J0Q>
5140 :       N$=""                                      <BK9>
5150 :       FOR J=0 TO 15                              <8NH>
5160 :        GET #8,A$: N$=N$+A$                       <UUK>
5170 :       NEXT J                                     <0Q5>
5180 :       IF ASC(MID$(N$,16,1))=191 THEN BEGIN
                                                        <FI4>
5190 :        CHAR ,50,12,STR$(Z+1)                     <UKG>
5200 :        KA$(Z)="": KA$(Z)=N$: Z=Z+1               <S3U>
5210 :       BEND                                       <38U>
5220 :      BEND                                        <RBU>
5230 :                                                  <AHB>
5240 :     BEND                                         <BDU>
5250 :                                                  <9H7>
5260 :    LOOP WHILE D<>-1                              <J66>
5270 :                                                  <8H3>
5280 :   LOOP WHILE NS<>255                             <RA3>
5290 :                                                  <FHV>
5300 :   CLOSE 8: CLOSE 1                               <63H>
5310 :                                                  <AHB>
5320 : REM - DIRECTORY AUFLISTEN                        <B1D>
5330 :                                                  <BHF>
5340 :   LZ=Z-1                                         <QT5>
5350 :   CHAR ,29,12,"DATEINAME:"+LEFT$(BL$,17)         <II1>
5360 :   IF Z=0 THEN CHAR ,29,12," KEINE DATEI VO
     RHANDEN !! "                                       <0JH>
5370 :                                                  <9H7>
5380 :   Z=0: A=0                                       <IIV>
5390 :   DO                                             <2OF>
5400 :    M=0                                           <7GN>
5410 :    CHAR ,40,12,LEFT$(KA$(Z),15)                  <FPP>
5420 :                                                  <C1H>
5430 :    SYS AM: PO=FN PO(0)                           <R02>
5440 :                                                  <F1T>
5450 : REM - EINTRAG HOCH                               <FHV>
5460 :                                                  <E1P>
5470 :    IF (PO>1229 AND PO<1239) OR (PO>1309 AN
     D PO<1319) THEN BEGIN                              <T35>
5480 :     M=1: Z=Z+1: IF Z>LZ THEN Z=LZ                <NP3>
5490 :    BEND                                          <3CU>
5500 :                                                  <C1H>
5510 : REM - EINTRAG RUNTER                             <CHJ>
5520 :                                                  <D1L>
5530 :    IF (PO>1239 AND PO<1249) OR (PO>1319 AN
     D PO<1329) THEN BEGIN                              <2CM>
5540 :     M=1: Z=Z-1: IF Z<0 THEN Z=0                  <MST>
5550 :    BEND                                          <J2U>
5560 :                                                  <F1T>
5570 : REM - EINTRAG OK                                 <FHV>
5580 :                                                  <811>
5590 :    IF (PO>1251 AND PO<1257) OR (PO>1331 AN
     D PO<1337) THEN BEGIN                              <LIL>
5600 :     DN$=KA$(Z)                                   <3CI>
5610 :    BEND                                          <30U>
5620 :                                                  <E1P>
5630 :   LOOP WHILE M=1                                 <Q09>
5640 :                                                  <D1L>
5650 :  BEND                                            <34U>
5660 :  CLOSE 8: CLOSE 1                                <6RI>
5670 :                                                  <CHJ>
5680 : REM - KATALOG-ENDE                               <B1D>
5690 :                                                  <BHF>
5700 : BEND                                             <R3U>
5710 :                                                  <EHR>
5720 : REM - LADEN/SPEICHERN/LOESCHEN                   <F1T>
5730 :                                                  <FHV>
5740 : M=0                                              <FHN>
5750 : IF (PO>333 AND PO<346) OR (PO>413 AND PO<4
     26) THEN M=1: A$="LADEN{4SPACE}"                   <M0T>
5760 : IF (PO>493 AND PO<506) OR (PO>573 AND PO<5
     86) THEN M=2: A$="SPEICHERN"                       <8JN>
5770 : IF (PO>653 AND PO<666) OR (PO>733 AND PO<7
     46) THEN M=3: A$="LÖSCHEN{2SPACE}"                 <LC9>
5780 :                                                  <A19>
5790 : IF M>0 THEN BEGIN                                <HN9>
5800 :  WINDOW 26,8,56,15,1                             <5U0>
5810 :   PRINT "Z++++++++++++++++++++++++++++++N"      <9R0>
5820 :   PRINT "K ";A$;LEFT$(BL$,19);"K"                <H6I>
5830 :   PRINT "G++++++++++++++++++++++++++++++M"      <Q3E>
5840 :   PRINT "K{29SPACE}K"                            <24D>
5850 :   PRINT "K DATEINAME:{18SPACE}K"                 <SR9>
5860 :   PRINT "K{29SPACE}K"                            <2CD>
5870 :   PRINT "K ESC = ENDE{9SPACE}RET = OK K"         <OOL>
5880 :   PRINT "I++++++++++++++++++++++++++++++O"      <BRC>
5890 :  PRINT WO$                                       <302>
5900 :                                                  <001>
5910 :  N$=LEFT$(DN$,15)                                <JP7>
5920 :  CHAR ,39,12,N$                                  <4A7>
5930 :                                                  <1G7>
5940 : REM - DATEINAME EINGEBEN                         <209>
5950 :                                                  <2GB>
5960 :  WINDOW 39,12,53,12,0                            <IQG>
5970 :   L=14: S=0                                      <ORQ>
5980 :   POKE VD,10: POKE VD+1,PEEK(VD+1) AND 159       <H15>
5990 :                                                  <4GJ>
6000 : REM - EINGABESCHLEIFE                            <F1T>
6010 :                                                  <FHV>
6020 :   DO                                             <0O7>
6030 :    GET KEY A$: A=ASC(A$)                         <OM8>
6040 :                                                  <D1L>
6050 : REM - DELETE                                     <DHN>
6060 :                                                  <C1H>
6070 :    IF A=20 THEN BEGIN                            <R9I>
6080 :     PRINT : N$=LEFT$(BL$,15): PRINT N$: S=
     0                                                  <5BD>
6090 :    BEND                                          <38U>
6100 :                                                  <E1P>
6110 : REM - CURSOR RECHTS                              <EHR>
6120 :                                                  <F1T>
6130 :    IF A=29 THEN BEGIN                            <OHK>
6140 :     IF S<L THEN PRINT A$;                        <RSM>
6150 :     S=S+1: IF S>L THEN S=L                       <81U>
6160 :    BEND                                          <B5U>
6170 :                                                  <DHN>
6180 :    IF A=34 OR A=64 THEN A=0                      <VO2>
6190 :                                                  <AHB>
6200 : REM - CURSOR LINKS                               <D1L>
6210 :                                                  <DHN>
6220 :    IF A=157 THEN BEGIN                           <N8A>
6230 :     PRINT A$;: S=S-1: IF S<0 THEN S=0            <URD>
6240 :    BEND                                          <B1U>
6250 :                                                  <FHV>
6260 : REM - ZEICHEN UEBERNEHMEN                        <E1P>
6270 :                                                  <EHR>
6280 :    IF A>31 AND A<128 THEN BEGIN                  <FVT>
6290 :     PRINT A$;: MID$(N$,S+1,1)=A$                 <I0B>
6300 :     S=S+1: IF S>L THEN S=L                       <A1M>
6310 :    BEND                                          <J6U>
6320 :                                                  <D1L>
6330 :   LOOP WHILE A<>13 AND A<>27                     <3N7>
6340 :                                                  <E1P>
6350 :   POKE VD,10: POKE VD+1,PEEK(VD+1) OR 32         <MC7>
6360 :  PRINT WO$                                       <9G2>
6370 :                                                  <FHV>
6380 : REM - EINGABEENDE                                <811>
6390 :                                                  <8H3>
6400 : BEND                                             <B9U>
6410 :                                                  <BHF>
6420 : REM - DISKETTENBEFEHL AUSFUEHREN                 <A19>
6430 :                                                  <AHB>
6440 : IF A=13 AND N$<>LEFT$(BL$,15) THEN BEGIN         <OA3>
6450 :                                                  <9H7>
6460 :  DN$=N$+CHR$(191)                                <O78>
6470 :  AD=8192+AS*4096                                 <8GE>
6480 :                                                  <F1T>
6490 : REM - ZEICHENSATZ LADEN                          <FHV>
6500 :                                                  <A19>
6510 :  IF M=1 THEN BEGIN                               <GF9>
6520 :   OPEN 8,8,8,DN$+",P,R": GOSUB 11960: CLOS
     E 8                                                <LT6>
6530 :   IF FF=0 THEN BLOAD (DN$),D0,U8,ON B0,P(A
     D): GOSUB 11960                                    <LT3>
6540 :   SYS TS,1,AS,0                                  <C1L>
6550 :  BEND                                            <JEU>
6560 :                                                  <915>
6570 : REM - ZEICHENSATZ SPEICHERN                      <9H7>
6580 :                                                  <E1P>
6590 :  IF M=2 THEN BEGIN                               <HND>
6600 :   SYS TS,1,AS,1                                  <S5L>
6610 :   BSAVE (DN$),D0,U8,ON B0,P(AD) TO P(AD+20
     48)                                                <5DR>
6620 :   GOSUB 11960                                    <AQK>
6630 :  BEND                                            <JEU>
6640 :                                                  <B1D>
6650 : REM - ZEICHENSATZ LOESCHEN                       <BHF>
6660 :                                                  <A19>
6670 :  IF M=3 THEN BEGIN                               <GV9>
6680 :   SCRATCH (DN$),D0,U8                            <JLD>
6690 :   GOSUB 11960                                    <SQK>
6700 :  BEND                                            <RFU>
6710 :                                                  <8H3>
6720 : REM - ENDE L/S/L                                 <915>
6730 :                                                  <9H7>
6740 : BEND                                             <RBU>
6750 : SYS SS,0                                         <BTM>
6760 : IF M=1 AND A=13 AND FF=0 THEN GOSUB 11850        <2HM>
6770 :                                                  <BHF>
6780 : REM - ENDE DATEIMENUE                            <C1H>
6790 :                                                  <CHJ>
6800 BEND                                               <PVE>
6810 :                                                  <7GV>
6820 : REM  ============================                <60P>
6830 : REM  WAHL                                        <6GR>
6840 : REM  ============================                <50L>
6850 :                                                  <5GN>
```

```
6860 IF PO>20 AND PO<25 THEN BEGIN                          <EAC>
6870 :                                                       <4GJ>
6880 : M=0                                                   <NUN>
6890 : CHAR ,20,0," WAHL ",1                                 <UVR>
6900 :                                                       <60P>
6910 : WINDOW 20,1,41,7,1                                    <J33>
6920 :  PRINT "K{20SPACE}K"                                  <S8F>
6930 :  PRINT "K ZEICHEN WFHLEN{5SPACE}K"                    <9IR>
6940 :  PRINT "K++++++++++++++++++++K"                       <ETL>
6950 :  PRINT "K ZEICHENSATZ NR.{2SPACE}0 K"                 <L6U>
6960 :  PRINT "K--------------------K"                       <PN7>
6970 :  PRINT "K ZEICHENSATZ NR.{2SPACE}1 K"                 <K2Q>
6980 :  PRINT "T++++++++++++++++++++@"                       <NEJ>
6990 : PRINT WO$                                             <402>
7000 :                                                       <D1L>
7010 : SYS AM : PO=FN PO(0)                                  <Q06>
7020 :                                                       <C1H>
7030 : REM - ZEICHEN WAEHLEN                                 <CHJ>
7040 :                                                       <F1T>
7050 : IF (PO>180 AND PO<201) OR (PO>260 AND PO<2
    81) THEN BEGIN                                          <4KV>
7060 :  Z=AZ: A2=143                                        <1VI>
7070 :  WINDOW 22,1,39,11,1                                  <MS0>
7080 :   PRINT "K{16SPACE}K"                                 <077>
7090 :   PRINT "K ZEICHEN HOCH{3SPACE}K"                     <57A>
7100 :   PRINT "K----------------K"                          <I40>
7110 :   PRINT "K ZEICHEN LINKS{2SPACE}K"                    <DB2>
7120 :   PRINT "K----------------K"                          <G40>
7130 :   PRINT "K ZEICHEN RECHTS K"                          <L86>
7140 :   PRINT "K----------------K"                          <M40>
7150 :   PRINT "K ZEICHEN RUNTER K"                          <9FA>
7160 :   PRINT "K++++++++++++++++K"                          <K40>
7170 :   PRINT "K OK{13SPACE}K"                              <EEA>
7180 :   PRINT "T++++++++++++++++@"                          <QSR>
7190 :  PRINT WO$                                            <E02>
7200 :                                                       <F1T>
7210 :  DO                                                   <6OV>
7220 :   X=0                                                 <L2T>
7230 :   SYS AM: PO=FN PO(0)                                 <P0A>
7240 :   IF Z=AZ THEN A1=143: ELSE A1=128                    <9N4>
7250 :                                                       <DHN>
7260 : REM - ZEICHEN HOCH                                    <C1H>
7270 :                                                       <CHJ>
7280 :   IF (PO>182 AND PO<200) OR (PO>262 AND PO
    <280) THEN BEGIN                                        <GEU>
7290 :    AT=A1: GOSUB 12210                                 <ARP>
7300 :    X=1: Z=Z-32: IF Z<0 THEN Z=Z+256                   <3KC>
7310 :    AT=A2: GOSUB 12210                                 <CRD>
7320 :   BEND                                                <B1U>
7330 :                                                       <FHV>
7340 : REM - ZEICHEN LINKS                                   <C1H>
7350 :                                                       <CHJ>
7360 :   IF (PO>342 AND PO<360) OR (PO>422 AND PO
    <440) THEN BEGIN                                        <3OI>
7370 :    AT=A1: GOSUB 12210                                 <CR1>
7380 :    X=1: Z=Z-1: IF Z<0 THEN Z=255                      <4CS>
7390 :    AT=A2: GOSUB 12210                                 <8RT>
7400 :   BEND                                                <BDU>
7410 :                                                       <9H7>
7420 : REM - ZEICHEN RECHTS                                  <811>
7430 :                                                       <8H3>
7440 :   IF (PO>502 AND PO<520) OR (PO>582 AND PO
    <600) THEN BEGIN                                        <G2V>
7450 :    AT=A1: GOSUB 12210                                 <ARP>
7460 :    X=1: Z=Z+1: IF Z>255 THEN Z=0                      <LJH>
7470 :    AT=A2: GOSUB 12210                                 <8RT>
7480 :   BEND                                                <B5U>
7490 :                                                       <DHN>
7500 : REM - ZEICHEN RUNTER                                  <811>
7510 :                                                       <8H3>
7520 :   IF (PO>662 AND PO<680) OR (PO>742 AND PO
    <760) THEN BEGIN                                        <5CQ>
7530 :    AT=A1: GOSUB 12210                                 <8RH>
7540 :    X=1: Z=Z+32: IF Z>255 THEN Z=Z-256                 <LFK>
7550 :    AT=A2: GOSUB 12210                                 <8RT>
7560 :   BEND                                                <B9U>
7570 :                                                       <BHF>
7580 : REM - ZEICHEN OK                                      <C1H>
7590 :                                                       <CHJ>
7600 :   IF (PO>822 AND PO<840) OR (PO>902 AND PO
    <920) THEN M=1: X=0                                     <HF4>
7610 :                                                       <BHF>
7620 :  LOOP WHILE X=1                                       <O12>
7630 :  PO=0                                                 <M2B>
7640 : BEND                                                  <BDU>
7650 :                                                       <9H7>
7660 : REM - ZEICHENSATZ AENDERN                             <811>
7670 :                                                       <8H3>
7680 : IF (PO>340 AND PO<361) OR (PO>420 AND PO<4
    41) THEN M=2: Z=0                                       <I3J>
7690 : IF (PO>500 AND PO<521) OR (PO>580 AND PO<6
    01) THEN M=2: Z=1                                       <GFJ>
7700 :                                                       <A19>
7710 : SYS SS,0                                              <BTM>
7720 :                                                       <B1D>
7730 : REM - NEUES ZEICHEN DARSTELLEN                        <BHF>
7740 :                                                       <811>
7750 : IF M=1 AND Z<>AZ THEN BEGIN                           <704>
7760 :  H=Z: Z=AZ: AT=128: GOSUB 12210                      <ONR>
7770 :  Z=H: AZ=Z: AT=143: GOSUB 12210                      <6O3>
7780 :  GOSUB 11830                                         <J2J>
7790 : BEND                                                  <J2U>
7800 :                                                       <50L>
7810 : REM - NEUER Z-SATZ DARSTELLEN                         <5GN>
7820 :                                                       <40H>
7830 : IF M=2 AND AS<>Z THEN BEGIN                           <BO5>
7840 :  SYS TS,1,AS,1                                       <SPM>
7850 :  SYS TS,1,Z,0                                        <8KG>
7860 :  SYS VP,234,3,48+Z                                   <BNM>
7870 :  AS=Z: GOSUB 11830                                   <I2N>
7880 : BEND                                                  <BTU>
7890 :                                                       <1G7>
7900 : REM - ENDE WAHLMENUE                                  <40H>
7910 :                                                       <4GJ>
7920 BEND                                                    <9TE>
7930 :                                                       <5GN>
7940 : REM  ===========================                      <60P>
7950 : REM  KOPIE                                            <6GR>
7960 : REM  ===========================                      <70T>
7970 :                                                       <7GV>
7980 IF PO>26 AND PO<32 THEN BEGIN                           <CIA>
7990 :                                                       <0G3>
8000 : M=0                                                   <NEN>
8010 : CHAR ,26,0," KOPIE ",1                                <MOV>
8020 :                                                       <I29>
8030 : Z=KZ: AT=159: GOSUB 12210                             <SUN>
8040 : A2=AT                                                 <MRH>
8050 :                                                       <HI7>
8060 : WINDOW 26,1,43,13,1                                   <VHC>
8070 :  PRINT "K{16SPACE}K"                                  <B37>
8080 :  PRINT "K ZEICHEN HOCH{3SPACE}K"                      <O3A>
8090 :  PRINT "K----------------K"                           <50O>
8100 :  PRINT "K ZEICHEN LINKS{2SPACE}K"                     <GF2>
8110 :  PRINT "K----------------K"                           <F0O>
8120 :  PRINT "K ZEICHEN RECHTS K"                           <8C6>
8130 :  PRINT "K----------------K"                           <D0O>
8140 :  PRINT "K ZEICHEN RUNTER K"                           <KBA>
8150 :  PRINT "K++++++++++++++++K"                           <B0O>
8160 :  PRINT "K KOPIEREN{7SPACE}K"                          <463>
8170 :  PRINT "K++++++++++++++++K"                           <90O>
8180 :  PRINT "K OK{13SPACE}K"                               <TAA>
8190 :  PRINT "T++++++++++++++++@"                           <7OR>
8200 : PRINT WO$                                             <NG2>
8210 :                                                       <HI7>
8220 : DO                                                    <ON7>
8230 :  X=0                                                  <1SD>
8240 :  SYS AM: PO=FN PO(0)                                  <4JS>
8250 :  IF Z=AZ THEN A1=143: ELSE A1=128                     <TB7>
8260 :                                                       <I29>
8270 : REM - ZEICHEN HOCH                                    <IIB>
8280 :                                                       <L2L>
8290 :  IF (PO>186 AND PO<204) OR (PO>266 AND PO<
    284) THEN BEGIN                                         <93J>
8300 :   AT=A1: GOSUB 12210                                  <H8N>
8310 :   X=1: Z=Z-32: IF Z<0 THEN Z=Z+256                    <N8F>
8320 :   AT=A2: GOSUB 12210                                  <J8J>
8330 :  BEND                                                 <2SU>
8340 :                                                       <I29>
8350 : REM - ZEICHEN LINKS                                   <IIB>
8360 :                                                       <J2D>
8370 :  IF (PO>346 AND PO<364) OR (PO>426 AND PO<
    444) THEN BEGIN                                         <QLV>
8380 :   AT=A1: GOSUB 12210                                  <L87>
8390 :   X=1: Z=Z-1: IF Z<0 THEN Z=255                       <QSS>
8400 :   AT=A2: GOSUB 12210                                  <L8B>
8410 :  BEND                                                 <2GU>
8420 :                                                       <M2P>
8430 : REM - ZEICHEN RECHTS                                  <MIR>
8440 :                                                       <L2L>
8450 :  IF (PO>506 AND PO<524) OR (PO>586 AND PO<
    604) THEN BEGIN                                         <9FI>
8460 :   AT=A1: GOSUB 12210                                  <L87>
8470 :   X=1: Z=Z+1: IF Z>255 THEN Z=0                       <B3H>
8480 :   AT=A2: GOSUB 12210                                  <H8R>
8490 :  BEND                                                 <2OU>
8500 :                                                       <M2P>
8510 : REM - ZEICHEN RUNTER                                  <MIR>
8520 :                                                       <N2T>
8530 :  IF (PO>666 AND PO<684) OR (PO>746 AND PO<
    764) THEN BEGIN                                         <S1N>
8540 :   AT=A1: GOSUB 12210                                  <L87>
8550 :   X=1: Z=Z+32: IF Z>255 THEN Z=Z-256                  <6NR>
8560 :   AT=A2: GOSUB 12210                                  <N83>
8570 :  BEND                                                 <2KU>
8580 :                                                       <I29>
8590 : REM - MUSTER KOPIEREN                                 <IIB>
8600 :                                                       <L2L>
8610 :  IF (PO>826 AND PO<844) OR (PO>906 AND PO<
    924) THEN BEGIN                                         <K7H>
8620 :   X=1: KZ=Z: SYS MC,1,AZ,0: SYS MC,1,KZ,1             <L1L>
8630 :  BEND                                                 <IMU>
8640 :                                                       <N2T>
8650 : REM - KOPIEREN OK                                     <NIV>
8660 :                                                       <M2P>
8670 :  IF (PO>986 AND PO<1004) OR (PO>1066 AND P
    O<1084) THEN X=0                                        <EQA>
8680 :                                                       <H25>
8690 : LOOP WHILE X=1                                        <1L4>
8700 : SYS SS,0                                              <4LM>
8710 :                                                       <KIJ>
8720 : REM - ENDE KOPIERMENUE                                <L2L>
8730 :                                                       <LIN>
8740 BEND                                                    <HEE>
8750 :                                                       <MIR>
8760 : REM  ===========================                      <N2T>
8770 : REM  ZEICHEN                                          <NIV>
8780 : REM  ===========================                      <G21>
8790 :                                                       <GI3>
```

Listing 1. (Fortsetzung)

```
8800 IF PO>33 AND PO<41 THEN BEGIN                   <3ID>
8810 :                                               <RJF>
8820 : M=0                                           <V7N>
8830 : CHAR ,33,0," ZEICHEN ",1                      <R6C>
8840 :                                               <P35>
8850 : WINDOW 33,1,52,21,1                           <K4C>
8860 :  PRINT "Ꝁ{18SPACE}Ꝁ"                           <4PC>
8870 :  PRINT "Ꝁ LÖSCHEN{10SPACE}Ꝁ"                   <3I1>
8880 :  PRINT "Ꝁ++++++++++++++++++Ꝁ"                 <P67>
8890 :  PRINT "Ꝁ INVERS{11SPACE}Ꝁ"                    <H1R>
8900 :  PRINT "Ꝁ++++++++++++++++++Ꝁ"                 <H27>
8910 :  PRINT "Ꝁ SENKR. SPIEGELN{2SPACE}Ꝁ"            <DET>
8920 :  PRINT "Ꝁ------------------Ꝁ"                 <NBT>
8930 :  PRINT "Ꝁ WAAGER. SPIEGELN Ꝁ"                  <P99>
8940 :  PRINT "Ꝁ++++++++++++++++++Ꝁ"                 <127>
8950 :  PRINT "Ꝁ HOCH ROTIEREN{4SPACE}Ꝁ"              <05T>
8960 :  PRINT "Ꝁ------------------Ꝁ"                 <7BT>
8970 :  PRINT "Ꝁ LINKS ROTIEREN{3SPACE}Ꝁ"             <MNS>
8980 :  PRINT "Ꝁ------------------Ꝁ"                 <VFT>
8990 :  PRINT "Ꝁ RECHTS ROTIEREN{2SPACE}Ꝁ"            <DDE>
9000 :  PRINT "Ꝁ------------------Ꝁ"                 <73S>
9010 :  PRINT "Ꝁ RUNTER ROTIEREN{2SPACE}Ꝁ"            <TTS>
9020 :  PRINT "Ꝁ++++++++++++++++++Ꝁ"                 <1A6>
9030 :  PRINT "Ꝁ LINKS DREHEN{5SPACE}Ꝁ"               <K9T>
9040 :  PRINT "Ꝁ------------------Ꝁ"                 <N3S>
9050 :  PRINT "Ꝁ RECHTS DREHEN{4SPACE}Ꝁ"              <G8F>
9060 :  PRINT "Ŧ++++++++++++++++++@"                 <M2R>
9070 : PRINT WO$                                     <K02>
9080 :                                               <L2L>
9090 : SYS AM: PO=FN PO(0)                           <236>
9100 :                                               <G21>
9110 : REM - LINKS DREHEN                            <GI3>
9120 :                                               <H25>
9130 : IF (PO>1473 AND PO<1493) OR (PO>1553 AND P
     O<1573) THEN BEGIN                              <SM4>
9140 :  FOR I=0 TO 3                                 <QE4>
9150 :   FOR J=I TO 6-I                              <OQ3>
9160 :    A=ZF(I,J)                                  <EK0>
9170 :    ZF(I,J)=ZF(J,7-I)                          <CDO>
9180 :    ZF(J,7-I)=ZF(7-I,7-J)                      <BT1>
9190 :    ZF(7-I,7-J)=ZF(7-J,I)                      <T7O>
9200 :    ZF(7-J,I)=A                                <GU1>
9210 :   NEXT J                                      <RQ5>
9220 :  NEXT I                                       <BG4>
9230 :  M=1                                          <RB7>
9240 : BEND                                          <ATU>
9250 :                                               <HI7>
9260 : REM - RECHTS DREHEN                           <G21>
9270 :                                               <GI3>
9280 : IF (PO>1633 AND PO<1653) OR (PO>1713 AND P
     O<1733) THEN BEGIN                              <9GG>
9290 :  FOR I=0 TO 3                                 <HE4>
9300 :   FOR J=I TO 6-I                              <SQ3>
9310 :    A=ZF(I,J)                                  <DK0>
9320 :    ZF(I,J)=ZF(7-J,I)                          <7CI>
9330 :    ZF(7-J,I)=ZF(7-I,7-J)                      <RK3>
9340 :    ZF(7-I,7-J)=ZF(J,7-I)                      <7KM>
9350 :    ZF(J,7-I)=A                                <4RP>
9360 :   NEXT J                                      <BG5>
9370 :  NEXT I                                       <RU4>
9380 :  M=1                                          <VFN>
9390 : BEND                                          <IIU>
9400 :                                               <L2L>
9410 : REM - LINKS ROTIEREN                          <LIN>
9420 :                                               <K2H>
9430 : IF (PO>993 AND PO<1013) OR (PO>1073 AND PO
     <1093) THEN BEGIN                               <280>
9440 :  FOR I=0 TO 7                                 <GU6>
9450 :   A=ZF(I,0)                                   <P7Q>
9460 :   FOR J=1 TO 7                                <VVI>
9470 :    ZF(I,J-1)=ZF(I,J)                          <B49>
9480 :   NEXT J                                      <BG5>
9490 :   ZF(I,7)=A                                   <KV1>
9500 :  NEXT I                                       <B84>
9510 :  M=1                                          <BD7>
9520 : BEND                                          <ALU>
9530 :                                               <LIN>
9540 : REM - RECHTS ROTIEREN                         <M2P>
9550 :                                               <MIR>
9560 : IF (PO>1153 AND PO<1173) OR (PO>1233 AND P
     O<1253) THEN BEGIN                              <04M>
9570 :  FOR I=0 TO 7                                 <HU6>
9580 :   A=ZF(I,7)                                   <NVT>
9590 :   FOR J=6 TO 0 STEP -1                        <0NV>
9600 :    ZF(I,J+1)=ZF(I,J)                          <B45>
9610 :   NEXT J                                      <RA5>
9620 :   ZF(I,0)=A                                   <SVT>
9630 :  NEXT I                                       <R24>
9640 :  M=1                                          <7CN>
9650 : BEND                                          <2KU>
9660 :                                               <K2H>
9670 : REM - HOCH ROTIEREN                           <KIJ>
9680 :                                               <J2D>
9690 : IF (PO>833 AND PO<853) OR (PO>913 AND PO<9
     33) THEN BEGIN                                  <UG1>
9700 :  FOR J=0 TO 7                                 <UVM>
9710 :   A=ZF(0,J)                                   <SR4>
9720 :   FOR I=1 TO 7                                <HU2>
9730 :    ZF(I-1,J)=ZF(I,J)                          <E00>
9740 :   NEXT I                                      <B84>
9750 :   ZF(7,J)=A                                   <QFL>
9760 :  NEXT J                                       <B05>
9770 :  M=1                                          <3C7>
9780 : BEND                                          <QRU>
9790 :                                               <IIB>
9800 : REM - RUNTER ROTIEREN                         <P35>
```

```
9810 :                                               <PJ7>
9820 : IF (PO>1313 AND PO<1333) OR (PO>1393 AND P
     O<1413) THEN BEGIN                              <GFM>
9830 :  FOR J=0 TO 7                                 <3VM>
9840 :   A=ZF(7,J)                                   <0RO>
9850 :   FOR I=6 TO 0 STEP -1                        <679>
9860 :    ZF(I+1,J)=ZF(I,J)                          <P3C>
9870 :   NEXT I                                      <QI4>
9880 :   ZF(0,J)=A                                   <LC5>
9890 :  NEXT J                                       <Q25>
9900 :  M=1                                          <F1N>
9910 : BEND                                          <IEU>
9920 :                                               <P35>
9930 : REM - SENKRECHT SPIEGELN                      <PJ7>
9940 :                                               <Q39>
9950 : IF (PO>513 AND PO<533) OR (PO>593 AND PO<6
     13) THEN BEGIN                                  <78G>
9960 :  FOR I=0 TO 7                                 <8U6>
9970 :   FOR J=0 TO 3                                <5FK>
9980 :    A=ZF(I,J)                                  <GK0>
9990 :    ZF(I,J)=ZF(I,7-J)                          <UNJ>
10000 :    ZF(I,7-J)=A                               <D9F>
10010 :   NEXT J                                     <1A5>
10020 :  NEXT I                                      <H04>
10030 :  M=1                                         <TV7>
10040 : BEND                                         <7LU>
10050 :                                              <58N>
10060 : REM - WAAGERECHT SPIEGELN                    <40H>
10070 :                                              <48J>
10080 : IF (PO>673 AND PO<693) OR (PO>753 AND PO<
      773) THEN BEGIN                                <VJ2>
10090 :  FOR I=0 TO 3                                <OQ2>
10100 :   FOR J=0 TO 7                               <VBG>
10110 :    A=ZF(I,J)                                 <406>
10120 :    ZF(I,J)=ZF(7-I,J)                         <D1T>
10130 :    ZF(7-I,J)=A                               <TB7>
10140 :   NEXT J                                     <H45>
10150 :  NEXT I                                      <1A4>
10160 :  M=1                                         <1SN>
10170 : BEND                                         <FKU>
10180 :                                              <209>
10190 : REM - ZEICHEN UMKEHREN                       <28B>
10200 :                                              <50L>
10210 : IF (PO>353 AND PO<373) OR (PO>433 AND PO<
      453) THEN BEGIN                                <OFF>
10220 :  FOR I=0 TO 7                                <NA0>
10230 :   FOR J=0 TO 7                               <QBG>
10240 :    ZF(I,J)=XOR(ZF(I,J),1)                    <SS9>
10250 :   NEXT J                                     <1A5>
10260 :  NEXT I                                      <H04>
10270 :  M=1                                         <TV7>
10280 : BEND                                         <7TU>
10290 :                                              <187>
10300 : REM - ZEICHEN LOESCHEN                       <40H>
10310 :                                              <48J>
10320 : IF (PO>193 AND PO<213) OR (PO>273 AND PO<
      293) THEN BEGIN                                <8HE>
10330 :  FOR I=0 TO 7                                <KA0>
10340 :   FOR J=0 TO 7                               <VBG>
10350 :    ZF(I,J)=0                                 <81B>
10360 :   NEXT J                                     <H85>
10370 :  NEXT I                                      <164>
10380 :  M=1                                         <9PN>
10390 : BEND                                         <VUU>
10400 :                                              <30D>
10410 : SYS SS,0                                     <F1N>
10420 :                                              <209>
10430 : REM - AENDERUNGEN AUSFUEHREN                 <28B>
10440 :                                              <105>
10450 : IF M=1 THEN BEGIN                            <2H2>
10460 :  FOR I=0 TO 7                                <VA0>
10470 :   D=0                                        <9DL>
10480 :   FOR J=0 TO 7                               <TBG>
10490 :    CHAR ,54+(J*2),7+(I*2)," ",ZF(I,J)        <H0V>
10500 :    IF ZF(I,J)=1 THEN D=D+2↑(7-J)             <TFA>
10510 :   NEXT J                                     <1U5>
10520 :   POKE DEC("0C80")+I,D                       <0V5>
10530 :  NEXT I                                      <1M4>
10540 :  SYS MC,1,AZ,1                               <60L>
10550 : BEND                                         <VUU>
10560 :                                              <105>
10570 : REM - ENDE ZEICHENMENUE                      <187>
10580 :                                              <60P>
10590 BEND                                           <JUU>
10600 :                                              <105>
10610 : REM  ===========================             <187>
10620 : REM  ALT                                     <001>
10630 : REM  ===========================             <083>
10640 :                                              <30D>
10650 IF PO>42 AND PO<46 THEN BEGIN                  <DM3>
10660 :                                              <209>
10670 : CHAR ,42,0," ALT ",1                         <FGL>
10680 :                                              <50L>
10690 : WINDOW 42,1,58,5,1                           <017>
10700 :  PRINT "Ꝁ{15SPACE}Ꝁ"                         <7VO>
10710 :  PRINT "Ꝁ ALTES ZEICHEN Ꝁ"                   <0QO>
10720 :  PRINT "Ꝁ+++++++++++++++Ꝁ"                   <781>
10730 :  PRINT "Ꝁ ALTER Z.-SATZ Ꝁ"                   <PCG>
10740 :  PRINT "Ŧ+++++++++++++++@"                   <507>
10750 : PRINT WO$                                    <401>
10760 :                                              <30D>
10770 : SYS AM: PO=FN PO(0)                          <KPU>
10780 : SYS SS,0                                     <CPN>
10790 :                                              <48J>
10800 : REM - ALTES ZEICHEN                          <FPT>
10810 :                                              <F9V>
```

```
10820 : IF (PO>202 AND PO<219) OR (PO>282 AND PO<
      299) THEN BEGIN                                        <7P4>
10830 :  D=8192+(AZ*8)+(AS*4096)                             <P9M>
10840 :  FOR I=0 TO 7                                        <5A0>
10850 :   POKE DEC("0C80")+I,PEEK(D+I)                       <JNC>
10860 :  NEXT I                                              <G84>
10870 :  SYS MC,1,AZ,1                                       <EOM>
10880 :  GOSUB 11830                                         <42V>
10890 : BEND                                                 <F8U>
10900 :                                                      <EPP>
10910 : REM - ALTER ZEICHENSATZ                              <E9R>
10920 :                                                      <FPT>
10930 : IF (PO>362 AND PO<379) OR (PO>442 AND PO<
      459) THEN BEGIN                                        <7D7>
10940 :  SYS TS,1,AS,0                                       <084>
10950 :  GOSUB 11830                                         <Q2V>
10960 : BEND                                                 <75U>
10970 :                                                      <D9N>
10980 : REM - ENDE ALTMENUE                                  <AP9>
10990 :                                                      <A9B>
11000 BEND                                                   <FTE>
11010 :                                                      <58N>
11020 : REM  ============================                    <40H>
11030 : REM  DRUCK                                           <48J>
11040 : REM  ============================                    <70T>
11050 :                                                      <78V>
11060 IF PO>47 AND PO<53 THEN BEGIN                          <FU6>
11070 :                                                      <68R>
11080 : GOSUB 12300                                          <AE6>
11090 :                                                      <187>
11100 BEND                                                   <7SE>
11110 :                                                      <48J>
11120 : REM  ============================                    <50L>
11130 : REM  ENDE                                            <58N>
11140 : REM  ============================                    <60P>
11150 :                                                      <68R>
11160 IF PO>54 AND PO<59 THEN BEGIN                          <DM7>
11170 :                                                      <78V>
11180 : CHAR ,54,0," ENDE ",1                                <PJJ>
11190 :                                                      <083>
11200 : WINDOW 24,7,54,18,1                                  <GGQ>
11210 :  PRINT "£+++++++++++++++++++++++++++++N"             <OF0>
11220 :  PRINT "K ENDE{24SPACE}K"                            <C09>
11230 :  PRINT "G+++++++++++++++++++++++++++++M"             <BNE>
11240 :  PRINT "K{29SPACE}K"                                 <JGD>
11250 :  PRINT "K ACHTUNG{21SPACE}K"                         <4A9>
11260 :  PRINT "K -------{21SPACE}K"                         <4AD>
11270 :  PRINT "K ALLE NOCH NICHT GESICHERTEN K"             <CAT>
11280 :  PRINT "K DATEN WERDEN GELÖSCHT !{5SPACE}
      K"                                                     <42D>
11290 :  PRINT "K{22SPACE}£++++N K"                          <2TL>
11300 :  PRINT "K{22SPACE}K OK K K"                          <K6P>
11310 :  PRINT "K{22SPACE}T++++@ K"                          <CSD>
11320 :  PRINT "T+++++++++++++++++++++++++++++@"             <RFC>
11330 : PRINT WO$                                            <101>
11340 :                                                      <40H>
11350 : SYS AM: PO=FN PO(0)                                  <JP2>
11360 :                                                      <50L>
11370 : REM --- PROGRAMM BEENDEN                             <58N>
11380 :                                                      <209>
11390 : IF (PO>1327 AND PO<1333) OR (PO>1407 AND
      PO<1413) THEN BEGIN                                    <M7M>
11400 :  SCNCLR : PRINT CHR$(27)+"L"                         <90B>
11410 :  PRINT "BIS BALD !"                                  <OJD>
11420 :  END                                                 <VNE>
11430 : BEND                                                 <VUU>
11440 : SYS SS,0                                             <F9N>
11450 :                                                      <38F>
11460 : REM --- ENDE ENDEMENUE                               <209>
11470 :                                                      <28B>
11480 BEND                                                   <FTE>
11490 :                                                      <58N>
11500 : REM - EDITMODUS WIEDER ANZEIGEN                      <001>
11510 :                                                      <083>
11520 CHAR ,7,0," EDIT ",1                                   <C0T>
11530 :                                                      <187>
11540 : REM - ENDE WENN PO<80                                <209>
11550 :                                                      <28B>
11560 BEND                                                   <VRE>
11570 :                                                      <38F>
11580 :                                                      <40H>
11590 : REM ==============                                   <48J>
11600 : REM SCHLEIFEN-ENDE                                   <30D>
11610 : REM ==============                                   <38F>
11620 :                                                      <209>
11630 LOOP WHILE X=0                                         <GL0>
11640 :                                                      <105>
11650 : REM - PRG.ENDE WENN X<>0                             <187>
11660 :                                                      <001>
11670 END                                                    <5KU>
11680 :                                                      <70T>
11690 :                                                      <78V>
11700 : REM  ===========================                     <209>
11710 : REM  UNTERPROGRAMME                                  <28B>
11720 : REM  ===========================                     <30D>
11730 :                                                      <38F>
11740 :                                                      <001>
11750 : REM - SCREENCODE IN ASCII                            <083>
11760 :                                                      <105>
11770 SYS SA,C                                               <89B>
11780 RREG A: C$=CHR$(A)                                     <DTU>
11790 IF C=34 OR C=162 THEN C$=CHR$(34)+CHR$(20)+
      CHR$(34)                                               <CKU>
11800 IF C>127 THEN C$=CHR$(18)+C$+CHR$(146)                 <VP4>
11810 RETURN                                                 <167>
11820 :                                                      <CPH>
11830 : REM - BITMUSTER IN ZF() U EDITOR                     <C9J>
11840 :                                                      <FPT>
11850 SYS MC,1,AZ,0                                          <AD4>
11860 FOR I=0 TO 7                                           <NA6>
11870 : D=PEEK(DEC("0C80")+I)                                <0KS>
11880 : FOR J=0 TO 7                                         <1BG>
11890 :   P=2↑(7-J)                                          <5VS>
11900 :   IF (D AND P)=P THEN ZF(I,J)=1 : ELSE ZF
      (I,J)=0                                                <EAJ>
11910 :   CHAR ,54+(J*2),7+(I*2)," ",ZF(I,J)                 <727>
11920 : NEXT J                                               <G05>
11930 NEXT I                                                 <9C4>
11940 RETURN                                                 <H87>
11950 :                                                      <E9R>
11960 : REM - FLOPPY-FEHLER                                  <FPT>
11970 :                                                      <F9V>
11980 FF=0                                                   <GAH>
11990 IF DS<>0 THEN BEGIN                                    <T97>
12000 : PRINT CHR$(7)                                        <SPI>
12010 : FF=1 : FS=DS : FF$=LEFT$(DS$,24)                     <JB7>
12020 : WINDOW 51,2,78,10,1                                  <5IQ>
12030 :  PRINT "£++++++++++++++++++++++++++N"                <33C>
12040 :  PRINT "K FLOPPY-FEHLER:{11SPACE}K"                  <5KD>
12050 :  PRINT "K{26SPACE}K"                                 <HVV>
12060 :  PRINT "K{26SPACE}K"                                 <TUF>
12070 :  PRINT "K{26SPACE}K"                                 <PUV>
12080 :  PRINT "K £++++++++++++++++++++++N K"                <H79>
12090 :  PRINT "K K BEHEBEN & BESTÄTIGEN K K"                <N30>
12100 :  PRINT "K T++++++++++++++++++++++@ K"                <I3H>
12110 :  PRINT "T++++++++++++++++++++++++++@"                <KJ2>
12120 : PRINT WO$                                            <581>
12130 : CHAR ,53,5,FF$,1                                     <P9S>
12140 : DO                                                   <RV7>
12150 :  SYS AM: PO=FN PO(0)                                 <NPI>
12160 :  IF (PO>693 AND PO<717) OR (PO>773 AND PO
      <797) THEN EXIT                                        <MAE>
12170 : LOOP                                                 <65P>
12180 BEND                                                   <NUE>
12190 RETURN                                                 <0A7>
12200 :                                                      <105>
12210 : REM - ATTR.POKE IM ZEICHENSATZ                       <187>
12220 :                                                      <001>
12230 A=INT(Z/32): AD=((14+A)*80+2048)+11+(Z-(A*3
      2))                                                    <HV1>
12240 AH=INT(AD/256): AL=AD-(AH*256)                         <PKV>
12250 SYS VP,AL,AH,AT                                        <MAV>
12260 RETURN                                                 <G07>
12270 :                                                      <28B>
12280 : REM - DRUCKROUTINE                                   <50L>
12290 :                                                      <58N>
12300 : CHAR ,47,0," DRUCK ",1                               <OVE>
12310 :                                                      <083>
12320 : M=0                                                  <1SN>
12330 : WINDOW 47,1,71,5,1                                   <JBF>
12340 :  PRINT "K{23SPACE}K"                                 <Q0V>
12350 :  PRINT "K ALTES + NEUES ZEICHEN K"                   <UAG>
12360 :  PRINT "K+++++++++++++++++++++++K"                   <5B0>
12370 :  PRINT "K HARDCOPY ZEICHENSATZ{2SPACE}K"             <UMS>
12380 :  PRINT "T+++++++++++++++++++++++@"                   <3RP>
12390 : PRINT WO$                                            <201>
12400 :                                                      <70T>
12410 : SYS AM: PO=FN PO(0)                                  <GPE>
12420 :                                                      <60P>
12430 : IF (PO>207 AND PO<232) OR (PO>287 AND PO<
      312) THEN M=1: A$="ALT / NEU"                          <N87>
12440 : IF (PO>367 AND PO<392) OR (PO>447 AND PO<
      472) THEN M=2: A$="HARDCOPY "                          <SPA>
12450 :                                                      <58N>
12460 : IF M<>0 THEN BEGIN                                   <0D3>
12470 :                                                      <48J>
12480 :  WINDOW 26,6,53,14,1                                 <RLI>
12490 :   PRINT "£++++++++++++++++++++++++++N"               <B2C>
12500 :   PRINT "K ";A$;"{16SPACE}K"                         <HDN>
12510 :   PRINT "G++++++++++++++++++++++++++M"               <QF1>
12520 :   PRINT "K DRUCKER EINGESCHALTET{4SPACE}K
      "                                                      <I0B>
12530 :   PRINT "K UND PAPIER EINGESPANNT ? K"               <G2V>
12540 :   PRINT "K{19SPACE}£++++N K"                         <D3N>
12550 :   PRINT "K{19SPACE}K OK K K"                         <UKB>
12560 :   PRINT "K{19SPACE}T++++@ K"                         <ABF>
12570 :   PRINT "T++++++++++++++++++++++++++@"               <CK2>
12580 :  PRINT WO$                                           <481>
12590 :                                                      <28B>
12600 :  SYS AM: PO=FN PO(0)                                 <I94>
12610 :                                                      <58N>
12620 :  IF (PO>1006 AND PO<1012) OR (PO>1086 AND
      PO<1092) THEN BEGIN                                    <I9C>
12630 :                                                      <48J>
12640 : REM - GEGENUEBERSTELLUNG ZEICHEN                     <70T>
12650 :                                                      <78V>
12660 :   IF M=1 THEN BEGIN                                  <3D5>
12670 :    OPEN 4,4: OPEN 5,4,5                              <DKE>
12680 :    O=0: SP=0                                         <QBR>
12690 :    DO                                                <TVV>
12700 :     IF O=0 THEN PRINT#4,"NORMALE ZEICHEN"
      : ELSE PRINT#4,"REVERSE ZEICHEN"                       <DDE>
12710 :     PRINT#4                                          <HA0>
12720 :     FOR I=0 TO 127 : A$=""                           <J04>
12730 :      SYS SA,I: RREG J                                <ENH>
12740 :      SYS MC,1,I+O,0                                  <TBE>
12750 :                                                      <68R>
```

Listing 1. (Fortsetzung)

```
12760 : REM - BITMUSTER UMRECHNEN                          <70T>
12770 :                                                    <78V>
12780 :      FOR I1=0 TO 7                                 <5JC>
12790 :       A=0                                          <BTV>
12800 :       FOR I2=0 TO 7                                <5MS>
12810 :        D=PEEK(DEC("0C80")+I2): P=2↑(7-I1)          <E9I>
12820 :        IF (D AND P)=P THEN A=A+2↑(7-I2)            <38L>
12830 :       NEXT I2                                      <78G>
12840 :       A$=A$+CHR$(A)                                <TN3>
12850 :      NEXT I1                                       <7SH>
12860 :                                                    <BP1>
12870 : REM - ZEICHEN AUSGEBEN                             <893>
12880 :                                                    <FPT>
12890 :      PRINT#5,A$                                    <5CF>
12900 :      PRINT#4,SPC(SP);CHR$(J);" = ";CHR$(2
     54);CHR$(141);                                        <RBJ>
12910 :      SP=SP+10: IF SP=80 THEN SP=0: PRINT#
     4                                                     <SPD>
12920 :     NEXT I                                         <GK4>
12930 :     O=O+128: PRINT#4                               <TAU>
12940 :    LOOP WHILE O<256                                <LUT>
12950 :    PRINT#4                                         <NA0>
12960 :    CLOSE 5: CLOSE 4                                <E2I>
12970 :   BEND                                             <FCU>
12980 :                                                    <EPP>
12990 : REM - HARDCOPY ZEICHENSATZ                         <E9R>
13000 :                                                    <105>
13010 :   IF M=2 THEN BEGIN                                <292>
13020 :    OPEN 6,4,6: PRINT#6,CHR$(20): CLOSE 6           <P7D>
13030 :    OPEN 4,4: OPEN 5,4,5                            <1LU>
13040 :    PRINT#4,"HARDCOPY ZEICHENSATZ"                  <O43>
13050 :    PRINT#4                                         <IQ0>
13060 :    SP=0                                            <U2P>
13070 :    FOR I=0 TO 255                                  <017>
13080 :     A$=""                                          <CBH>
13090 :     SYS MC,1,I,0                                   <67E>
13100 :                                                    <0O1>
13110 : REM - BITMUSTER UMRECHNEN                          <083>
13120 :                                                    <105>
13130 :     FOR I1=0 TO 7                                  <3JC>
13140 :      A=0                                           <VVF>
13150 :      FOR I2=0 TO 7                                 <3IS>
13160 :       D=PEEK(DEC("0C80")+I2): P=2↑(7-I1)           <VBM>
13170 :       IF (D AND P)=P THEN A=A+2↑(7-I2)             <F0K>
13180 :      NEXT I2                                       <4SG>
13190 :      A$=A$+CHR$(A)                                 <C35>
13200 :     NEXT I1                                        <58H>
13210 :                                                    <38F>
13220 : REM - ZEICHEN AUSGEBEN                             <209>
13230 :                                                    <28B>
13240 :     PRINT#5,A$                                     <V8F>
13250 :     PRINT#4,SPC(SP);CHR$(254);CHR$(141);           <R1S>
13260 :     SP=SP+1: IF SP=32 THEN SP=0: PRINT#4           <84T>
13270 :    NEXT I                                          <1Q4>
13280 :    OPEN 6,4,6: PRINT#6,CHR$(31): CLOSE 6           <NAL>
13290 :    CLOSE 5: CLOSE 4                                <VQL>
13300 :   BEND                                             <NRU>
13310 :                                                    <28B>
13320 :  BEND                                              <7PU>
13330 :                                                    <38F>
13340 : REM --- ENDE WENN M<>0                             <0O1>
13350 :                                                    <083>
13360 : BEND                                               <7TU>
13370 : SYS SS,0                                           <E1N>
13380 :                                                    <6OP>
13390 : REM --- ENDE DRUCKMENUE                            <68R>
13400 :                                                    <5OL>
13410 RETURN                                               <067>
```

Listing 1. (Schluß)

```
Name : modichar.obj        0c00 0c8b
------------------------------------
0c00 : c9 01 f0 03 a9 20 2c a9   86
0c08 : 30 85 fb a9 00 85 fa e8   19
0c10 : ca f0 0d a5 fa 18 69 10   81
0c18 : 85 fa 90 f4 e6 fb d0 f0   50
0c20 : 84 fc a0 00 a9 12 a6 fb   08
0c28 : 20 5e 0c a9 13 a6 fa 20   42
0c30 : 5e 0c a5 fc c9 01 f0 0c   1e
0c38 : a9 1f 20 6f 0c 8a 99 80   e3
0c40 : 0c 18 90 09 b9 80 0c aa   c3
0c48 : a9 1f 20 5e 0c a5 fa 18   5f
0c50 : 69 01 85 fa 90 02 e6 fb   a7
0c58 : c8 c0 08 d0 c7 60 8d 00   52
0c60 : d6 2c 00 d6 10 fb 8e 01   44
0c68 : d6 2c 00 d6 10 fb 60 8d   ad
0c70 : 00 d6 2c 00 d6 10 fb ae   22
0c78 : 01 d6 2c 00 d6 10 fb 60   8e
0c80 : c0 f0 7c 7f 7c 36 03 01   4f
0c88 : 0a 00 a0 20 56 4f 4e 20   18
```

Listing 3. Modifiziert die Zeichen. Unterprogramm zu Charedit.

```
Name : vdc-poke.obj        0c90 0cbf
------------------------------------
0c90 : 85 fa 86 fb 84 fc a9 12   ae
0c98 : a6 fb 20 ac 0c a9 13 a6   81
0ca0 : fa 20 ac 0c a9 1f a6 fc   7f
0ca8 : 20 ac 0c 60 8d 00 d6 2c   ba
0cb0 : 00 d6 10 fb 8e 01 d6 2c   43
0cb8 : 00 d6 10 fb 60 00 20 41   b0
```

Listing 2. Beschreibt den VDC Speicher. Unterprogramm zu Charedit. Bitte mit dem MSE im C64-Modus eingeben. Ebenso Listings 2 bis 9.

```
Name : swapscrn.obj        0d00 0d93
------------------------------------
0d00 : c9 01 f0 0c a9 10 85 fb   31
0d08 : a9 00 85 fc 85 fa f0 0a   ba
0d10 : a9 10 85 fc a9 00 85 fb   6b
0d18 : 85 fa a0 00 a9 18 20 7f   1d
0d20 : 0d 86 fd 8a 09 80 aa a9   d4
0d28 : 18 20 6e 0d a9 12 a6 fc   4d
0d30 : 20 6e 0d a9 13 a6 fa 20   92
0d38 : 6e 0d a9 20 a6 fb 20 6e   43
0d40 : 0d a9 21 a6 fa 20 6e 0d   c3
0d48 : a9 1e a2 4f 20 6e 0d a5   88
0d50 : fa 18 69 4f 85 fa 90 04   15
0d58 : e6 fb e6 fc 2c 00 d6 10   d3
0d60 : fb c8 c0 33 d0 c6 a9 18   70
0d68 : a6 fd 20 6e 0d 60 8d 00   ed
0d70 : d6 2c 00 d6 10 fb 8e 01   54
0d78 : d6 2c 00 d6 10 fb 60 8d   bd
0d80 : 00 d6 2c 00 d6 10 fb ae   32
0d88 : 01 d6 2c 00 d6 10 fb 60   9e
0d90 : a0 d6 10 05 85 f8 4c 07   9f
```

Listing 5. Vertauscht zwei Bildschirme. Unterprogramm zu Charedit.

```
Name : transset.obj        0b00 0ba9
------------------------------------
0b00 : c9 01 f0 03 a9 20 2c a9   86
0b08 : 30 85 fb 18 69 10 85 fe   28
0b10 : e0 01 f0 03 a9 20 2c a9   ad
0b18 : 30 85 fd a9 00 85 fc 85   eb
0b20 : fa 8c fd 03 a0 00 8c fe   7a
0b28 : 03 a9 12 a6 fb 20 84 0b   42
0b30 : a9 13 a6 fa 20 84 0b ad   19
0b38 : fd 03 c9 01 f0 0b b1 fc   71
0b40 : aa a9 1f 20 84 0b 18 90   ad
0b48 : 08 a9 1f 20 95 0b 8a 91   f0
0b50 : fc a5 fc 18 69 01 85 fc   10
0b58 : 90 02 e6 fd a5 fa 18 69   c8
0b60 : 01 85 fa 90 02 e6 fb ae   99
0b68 : fe 03 e8 8e fe 03 e0 08   8f
0b70 : d0 b7 a5 fa 18 69 08 85   dc
0b78 : fa 90 02 e6 fb a5 fb c5   80
0b80 : fe d0 a1 60 8d 00 d6 2c   e7
0b88 : 00 d6 10 fb 8e 01 d6 2c   1b
0b90 : 00 d6 10 fb 60 8d 00 d6   9f
0b98 : 2c 00 d6 10 fb ae 01 d6   63
0ba0 : 2c 00 d6 10 fb 60 48 00   68
0ba8 : a0 01 d4 a2 d0 a0 0d 8e   b6
```

Listing 4. Verschiebt den Zeichensatz. Unterprogramm zu Charedit.

```
Name : charmove.obj        1300 14c8
------------------------------------
1300 : ac bf 14 b9 8d 14 18 6d   7d
1308 : c0 14 85 fb b9 a6 14 69   a7
1310 : 00 85 fa a9 12 a6 fa 20   49
1318 : 6b 14 a9 13 a6 fb 20 6b   fc
1320 : 14 a9 1f 20 7c 14 86 fc   51
1328 : a5 fa 18 69 08 85 fd aa   78
1330 : a9 12 20 6b 14 a9 13 a6   80
1338 : fb 20 6b 14 a9 1f 20 7c   ae
1340 : 14 86 fe a9 12 a6 fd 20   1b
1348 : 6b 14 a9 13 a6 fb 20 6b   2c
1350 : 14 a9 1f ae c2 14 20 6b   fb
1358 : 14 a9 12 a6 fa 20 6b 14   21
1360 : a9 13 a6 fb 20 6b 14 a9   bd
1368 : 1f ae c1 14 20 6b 14 ad   da
1370 : 00 dc 8d fd 03 29 10 8d   d6
1378 : ff 03 f0 62 ad fd 03 29   aa
1380 : 0f 8d fd 03 a9 0f 38 ed   05
1388 : fd 03 f0 e3 c9 01 d0 05   b1
1390 : 20 11 14 90 49 c9 02 d0   dd
1398 : 05 20 21 14 90 40 c9 04   b2
13a0 : d0 05 20 31 14 90 37 c9   57
13a8 : 05 d0 08 20 11 14 20 31   b0
13b0 : 14 90 2b c9 06 d0 08 20   58
13b8 : 21 14 20 31 14 90 1f c9   e7
13c0 : 08 d0 05 20 41 14 90 16   99
13c8 : c9 09 d0 08 20 11 14 20   66
13d0 : 41 14 90 0a c9 0a d0 97   e0
13d8 : 20 21 14 20 41 14 a9 12   11
13e0 : a6 fd 20 6b 14 a9 13 a6   23
13e8 : fb 20 6b 14 a9 1f a6 fe   7d
13f0 : 20 6b 14 a9 12 a6 fa 20   83
13f8 : 6b 14 a9 13 a6 fb 20 6b   dc
1400 : 14 a9 1f a6 fc 20 6b 14   2c
1408 : ad ff 03 f0 03 4c 00 13   4c
1410 : 60 ad bf 14 38 e9 01 c9   24
1418 : ff f0 03 8d bf 14 4c 51   72
1420 : 14 ad bf 14 18 69 01 c9   e2
1428 : 19 f0 03 8d bf 14 4c 51   9c
1430 : 14 ad c0 14 38 e9 01 c9   38
1438 : ff f0 03 8d c0 14 4c 5e   bc
1440 : 14 ad c0 14 18 69 01 c9   42
1448 : 50 f0 03 8d c0 14 4c 5e   1d
1450 : 14 a9 ff a8 88 d0 fd 38   c5
1458 : ed c3 14 b0 f7 60 a9 ff   6b
1460 : a8 88 d0 fd 38 ed c4 14   6e
1468 : b0 f7 60 8d 00 d6 2c 00   45
1470 : d6 10 fb 8e 01 d6 2c 00   96
1478 : d6 10 fb 60 8d 00 d6 2c   ee
1480 : 00 d6 10 fb ae 01 d6 2c   15
1488 : 00 d6 10 fb 60 00 50 a0   ff
1490 : f0 40 90 e0 30 80 d0 20   6b
1498 : 70 c0 10 60 b0 00 50 a0   06
14a0 : f0 40 90 e0 30 80 00 00   f8
14a8 : 00 00 01 01 01 02 02 02   35
14b0 : 03 03 03 04 04 04 05 05   f5
14b8 : 05 05 06 06 06 07 07 13   5d
14c0 : 2b 00 0f 02 05 53 00 2e   37
```

Listing 6. Bewegt ein Zeichen auf dem Bildschirm. Unterprogramm zu Charedit.

```
Name : scntoasc.obj        0cc0 0ce6
-------------------------------------
0cc0 : 29 7f c9 20 b0 04 a2 40   55
0cc8 : 90 12 c9 40 b0 04 a2 00   91
0cd0 : 90 0a c9 60 b0 04 a2 20   da
0cd8 : 90 02 a2 40 86 fa 18 65   85
0ce0 : fa 85 fa 60 41 2d 41 44   73
```

Listing 7. Wandelt Bildschirm- in ASCII-Code. Unterprogramm zu Charedit.

```
Name : fett (g/k)       ' 3000 3801
-------------------------------------
3000 : 3e 60 3c 66 3c 06 7c 00   30
3008 : 00 00 3c 06 3e 66 3e 00   e8
3010 : 60 60 7c 66 66 66 7c 00   18
3018 : 00 00 3c 66 60 66 3c 00   1e
3020 : 06 06 3e 66 66 66 3e 00   18
3028 : 00 00 3c 66 7e 60 3c 00   e0
3030 : 0e 18 7e 18 18 18 18 00   90
3038 : 00 00 3e 66 66 3e 06 3c   7e
3040 : 60 60 7c 66 66 66 66 00   ef
3048 : 18 00 78 18 18 18 7e 00   be
3050 : 0c 00 3c 0c 0c 0c 0c 78   2f
3058 : 60 60 66 6c 78 6c 66 00   94
3060 : 78 18 18 18 18 18 7e 00   2a
3068 : 00 00 fe db db db db 00   af
3070 : 00 00 7c 66 66 66 66 00   8f
3078 : 00 00 3c 66 66 66 3c 00   df
3080 : 00 00 7c 66 66 7c 60 60   f9
3088 : 00 00 3e 66 66 3e 06 06   61
3090 : 00 00 6e 70 60 60 60 00   c5
3098 : 00 00 3e 60 3c 06 7c 00   1a
30a0 : 18 18 7e 18 18 18 0e 00   e1
30a8 : 00 00 66 66 66 66 3e 00   a1
30b0 : 00 00 66 66 66 3c 18 00   bf
30b8 : 00 00 c3 db db db 7e 00   bb
30c0 : 00 00 66 3c 18 3c 66 00   de
30c8 : 00 00 66 66 66 3e 06 3c   18
30d0 : 00 00 7e 0c 18 30 7e 00   ee
30d8 : 3c 30 30 30 30 30 3c 00   b4
30e0 : c0 60 30 18 0c 06 03 00   dc
30e8 : 3c 0c 0c 0c 0c 0c 3c 00   c1
30f0 : 18 3c 7e 18 18 18 18 00   6c
30f8 : 00 00 00 00 00 00 00 ff   f8
3100 : 00 00 00 00 00 00 00 00   01
3108 : 18 18 18 18 18 00 18 00   17
3110 : 66 66 66 00 00 00 00 00   43
3118 : 66 66 ff 66 ff 66 66 00   4b
3120 : 18 3e 60 3c 06 7c 18 00   9c
3128 : 62 66 0c 18 30 66 46 00   13
3130 : 3c 66 3c 38 67 66 3f 00   5c
3138 : 18 18 18 00 00 00 00 00   63
3140 : 0c 18 30 30 30 18 0c 00   5e
3148 : 30 18 0c 0c 0c 18 30 00   4b
3150 : 08 2a 1c 3e 1c 2a 08 00   6f
3158 : 00 18 18 7e 18 18 00 00   7d
3160 : 00 00 00 00 00 18 18 30   e2
3168 : 00 00 00 7e 00 00 00 00   38
3170 : 00 00 00 00 00 18 18 00   92
3178 : 03 06 0c 18 30 60 c0 00   8e
3180 : 3c 66 6e 76 66 66 3c 00   e4
3188 : 18 18 38 18 18 18 7e 00   fa
3190 : 3c 66 06 0c 30 60 7e 00   02
3198 : 3c 66 06 1c 06 66 3c 00   91
31a0 : 06 0e 1e 66 7f 06 06 00   42
31a8 : 7e 60 7c 06 06 66 3c 00   bb
31b0 : 3c 66 60 7c 66 66 3c 00   52
31b8 : 7e 66 06 0c 18 18 18 00   0f
31c0 : 3c 66 66 3c 66 66 3c 00   db
31c8 : 3c 66 66 3e 06 66 3c 00   1d
31d0 : 00 00 18 00 00 18 00 00   97
31d8 : 00 00 18 00 00 18 18 30   60
31e0 : 0e 18 30 60 30 18 0e 00   0e
31e8 : 00 00 7e 00 7e 00 00 00   70
31f0 : 70 18 0c 06 0c 18 70 00   74
31f8 : 3c 66 06 0c 18 00 18 00   4c
3200 : 30 18 0c 00 00 00 00 00   40
3208 : 18 3c 66 7e 66 66 66 00   db
3210 : 7c 66 66 7c 66 66 7c 00   74
3218 : 3c 66 60 60 60 66 3c 00   d6
3220 : 78 6c 66 66 66 6c 78 00   e0
3228 : 7e 60 60 78 60 60 7e 00   00
3230 : 7e 60 60 78 60 60 60 00   90
3238 : 3c 66 60 6e 66 66 3c 00   18
3240 : 66 66 66 7e 66 66 66 00   76
3248 : 3c 18 18 18 18 18 3c 00   cd
3250 : 1e 0c 0c 0c 0c 6c 38 00   fe
3258 : 66 6c 78 70 78 6c 66 00   a5
3260 : 60 60 60 60 60 60 7e 00   17
3268 : 63 77 7f 6b 63 63 63 00   b3
3270 : 66 76 7e 7e 6e 66 66 00   34
3278 : 3c 66 66 66 66 66 3c 00   d8
3280 : 7c 66 66 7c 60 60 60 00   e3
3288 : 3c 66 66 66 66 3c 0e 00   de
3290 : 7c 66 66 7c 78 6c 66 00   ed
3298 : 3c 66 60 3c 06 66 3c 00   2c
32a0 : 7e 18 18 18 18 18 18 00   d6
32a8 : 66 66 66 66 66 66 3c 00   32
32b0 : 66 66 66 66 66 3c 18 00   58
32b8 : 63 63 63 6b 7f 77 63 00   54
32c0 : 66 66 3c 18 3c 66 66 00   fc
32c8 : 66 66 66 3c 18 18 18 00   25
32d0 : 7e 06 0c 18 30 60 7e 00   57
32d8 : 42 3c 66 7e 66 66 66 00   d5
32e0 : 42 3c 66 66 66 66 3c 00   31
32e8 : 18 66 66 66 66 66 3c 00   24
32f0 : 00 00 01 7e b6 36 36 00   f6
32f8 : 00 00 00 00 00 00 00 ff   f8
3300 : 00 00 00 00 00 00 00 00   01
3308 : 18 18 18 18 18 18 18 18   08
3310 : 18 18 18 1f 1f 00 00 00   10
3318 : 18 18 18 ff ff 00 00 00   42
3320 : 18 18 18 f8 f8 00 00 00   f9
3328 : 18 18 18 1f 1f 18 18 18   7a
3330 : 00 00 00 ff ff 00 00 00   30
3338 : 18 18 18 f8 f8 18 18 18   62
3340 : 00 00 00 1f 1f 18 18 18   68
3348 : 00 00 00 ff ff 18 18 18   9a
3350 : 00 00 00 f8 f8 18 18 18   50
3358 : 18 18 18 ff ff 18 18 18   d4
3360 : 0c 30 3c 66 7e 60 3c 00   3c
3368 : 1c 36 30 78 30 73 de 00   d5
3370 : 30 0c 3c 66 7e 60 3c 00   5e
3378 : 0c 18 30 00 00 00 00 00   9d
3380 : 3c 66 6e 6e 60 66 3c 00   83
3388 : 00 00 66 66 66 7e da 80   b5
3390 : 30 0c 3c 06 3e 66 3b 00   9a
3398 : 30 0c 66 66 66 66 3e 00   c7
33a0 : 18 24 3c 06 3e 66 3b 00   9e
33a8 : 18 24 3c 66 7e 60 3c 00   8a
33b0 : 18 24 78 18 18 18 7e 00   38
33b8 : 18 24 3c 66 66 66 3c 00   49
33c0 : 18 24 66 66 66 66 [illegible] 00   e3
33c8 : 1f 18 18 18 d8 38 [illegible] 00   ac
33d0 : ff 63 30 18 30 63 ff 00   ae
33d8 : 24 00 3c 06 3e 66 3e 00   dc
33e0 : 24 00 3c 66 66 66 3c 00   6b
33e8 : 24 00 66 66 66 66 3e 00   05
33f0 : 3c 66 66 6c 66 66 6c 60   93
33f8 : 18 24 00 00 00 00 00 00   23
3400 : c1 9f c3 99 c3 f9 83 ff   cf
3408 : ff ff c3 f9 c1 99 c1 ff   27
3410 : 9f 9f 83 99 99 99 83 ff   07
3418 : ff ff c3 99 9f 99 c3 ff   11
3420 : f9 f9 c1 99 99 99 c1 ff   27
3428 : ff ff c3 99 81 9f c3 ff   6f
3430 : f1 e7 81 e7 e7 e7 e7 ff   cf
3438 : ff ff c1 99 99 c1 f9 c3   f1
3440 : 9f 9f 83 99 99 99 99 ff   90
3448 : e7 ff 87 e7 e7 e7 81 ff   d1
3450 : f3 ff c3 f3 f3 f3 f3 87   70
3458 : 9f 9f 99 93 87 93 99 ff   1b
3460 : 87 e7 e7 e7 e7 e7 81 ff   95
3468 : ff ff 01 24 24 24 24 ff   20
3470 : ff ff 83 99 99 99 99 ff   50
3478 : ff ff c3 99 99 99 c3 ff   10
3480 : ff ff 83 99 99 83 9f 9f   06
3488 : ff ff c1 99 99 c1 f9 f9   ae
3490 : ff ff 91 8f 9f 9f 9f ff   5b
3498 : ff ff c1 9f c3 f9 83 ff   15
34a0 : e7 e7 81 e7 e7 e7 f1 ff   5e
34a8 : ff ff 99 99 99 99 c1 ff   ae
34b0 : ff ff 99 99 99 c3 e7 ff   a0
34b8 : ff ff 3c 24 24 24 81 ff   b4
34c0 : ff ff 99 c3 e7 c3 99 ff   a1
34c8 : ff ff 99 99 99 c1 f9 c3   77
34d0 : ff ff 81 f3 e7 cf 81 ff   b1
34d8 : c3 cf cf cf cf cf c3 ff   fb
34e0 : 3f 9f cf e7 f3 f9 fc ff   e3
34e8 : c3 f3 f3 f3 f3 f3 c3 ff   0e
34f0 : e7 c3 81 e7 e7 e7 e7 ff   73
34f8 : ff ff ff ff ff ff ff 00   f7
3500 : ff ff ff ff ff ff ff ff   ff
3508 : e7 e7 e7 e7 e7 ff e7 ff   f8
3510 : 99 99 99 ff ff ff ff ff   dc
3518 : 99 99 00 99 00 99 99 ff   e4
3520 : e7 c1 9f c3 f9 83 e7 ff   a3
3528 : 9d 99 f3 e7 cf 99 b9 ff   3c
3530 : c3 99 c3 c7 98 99 c0 ff   03
3538 : e7 e7 e7 ff ff ff ff ff   0d
3540 : f3 e7 cf cf cf e7 f3 ff   21
3548 : cf e7 f3 f3 f3 e7 cf ff   44
3550 : f7 d5 e3 c1 e3 d5 f7 ff   30
3558 : ff e7 e7 81 e7 e7 ff ff   33
3560 : ff ff ff ff ff e7 e7 cf   dd
3568 : ff ff ff 81 ff ff ff ff   97
3570 : ff ff ff ff ff e7 e7 ff   4d
3578 : fc f9 f3 e7 cf 9f 3f ff   62
3580 : c3 99 91 89 99 99 c3 ff   1b
3588 : e7 e7 c7 e7 e7 e7 81 ff   15
3590 : c3 99 f9 f3 cf 9f 81 ff   1d
3598 : c3 99 f9 e3 f9 99 c3 ff   9e
35a0 : f9 f1 e1 99 80 f9 f9 ff   fd
35a8 : 81 9f 83 f9 f9 99 c3 ff   94
35b0 : c3 99 9f 83 99 99 c3 ff   0e
35b8 : 81 99 f9 f3 e7 e7 e7 ff   60
35c0 : c3 99 99 c3 99 99 c3 ff   a4
35c8 : c3 99 99 c1 f9 99 c3 ff   72
35d0 : ff ff e7 ff ff e7 ff ff   08
35d8 : ff ff e7 ff ff e7 e7 cf   4f
35e0 : f1 e7 cf 9f cf e7 f1 ff   b1
35e8 : ff ff 81 ff 81 ff ff ff   5f
35f0 : 8f e7 f3 f9 f3 e7 8f ff   6c
35f8 : c3 99 f9 f3 e7 ff e7 ff   a3
3600 : cf e7 f3 ff ff ff ff ff   c0
3608 : e7 c3 99 81 99 99 99 ff   34
3610 : 83 99 99 83 99 99 83 ff   ab
3618 : c3 99 9f 9f 9f 99 c3 ff   59
3620 : 87 93 99 99 99 93 87 ff   5f
3628 : 81 9f 9f 87 9f 9f 81 ff   4f
3630 : 81 9f 9f 87 9f 9f 9f ff   cf
3638 : c3 99 9f 91 99 99 c3 ff   57
3640 : 99 99 99 81 99 99 99 ff   09
3648 : c3 e7 e7 e7 e7 e7 c3 ff   c2
3650 : e1 f3 f3 f3 f3 93 c7 ff   a1
3658 : 99 93 87 8f 87 93 99 ff   0a
3660 : 9f 9f 9f 9f 9f 9f 81 ff   a8
3668 : 9c 88 80 94 9c 9c 9c ff   1c
3670 : 99 89 81 81 91 99 99 ff   ab
3678 : c3 99 99 99 99 99 c3 ff   17
3680 : 83 99 99 83 9f 9f 9f ff   1c
3688 : c3 99 99 99 99 c3 f1 ff   31
3690 : 83 99 99 83 87 93 99 ff   32
3698 : c3 99 9f c3 f9 99 c3 ff   04
36a0 : 81 e7 e7 e7 e7 e7 e7 ff   69
36a8 : 99 99 99 99 99 99 c3 ff   1d
36b0 : 99 99 99 99 99 c3 e7 ff   07
36b8 : 9c 9c 9c 94 80 88 9c ff   1b
36c0 : 99 99 c3 e7 c3 99 99 ff   83
36c8 : 99 99 99 c3 e7 e7 e7 ff   6a
36d0 : 81 f9 f3 e7 cf 9f 81 ff   48
36d8 : bd c3 99 81 99 99 99 ff   da
36e0 : bd c3 99 99 99 99 c3 ff   8e
36e8 : e7 99 99 99 99 99 c3 ff   ab
36f0 : ff ff fe 81 49 c9 c9 ff   e9
36f8 : ff ff ff ff ff ff ff 00   f7
3700 : ff ff ff ff ff ff ff ff   ff
3708 : e7 e7 e7 e7 e7 e7 e7 e7   07
3710 : e7 e7 e7 e0 e0 ff ff ff   0f
3718 : e7 e7 e7 00 00 ff ff ff   ed
3720 : e7 e7 e7 07 07 ff ff ff   46
3728 : e7 e7 e7 e0 e0 e7 e7 e7   d5
3730 : ff ff ff 00 00 ff ff ff   2f
3738 : e7 e7 e7 07 07 e7 e7 e7   0d
3740 : ff ff ff e0 e0 e7 e7 e7   17
3748 : ff ff ff 00 00 e7 e7 e7   f5
3750 : ff ff ff 07 07 e7 e7 e7   4f
3758 : e7 e7 e7 00 00 e7 e7 e7   db
3760 : f3 cf c3 99 81 9f c3 ff   83
3768 : e3 c9 cf 87 cf 8c 21 ff   fa
3770 : cf f3 c3 99 81 9f c3 ff   81
3778 : f3 e7 cf ff ff ff ff ff   53
3780 : c3 99 91 91 9f 99 c3 ff   7c
3788 : ff ff 99 99 99 81 25 7f   5a
3790 : cf f3 c3 f9 c1 99 c4 ff   85
3798 : cf f3 99 99 99 99 c1 ff   68
37a0 : e7 db c3 f9 c1 99 c4 ff   a1
37a8 : e7 db c3 99 81 9f c3 ff   c5
37b0 : e7 db 87 e7 e7 e7 81 ff   27
37b8 : e7 db c3 99 99 99 c3 ff   26
37c0 : e7 db 99 99 99 99 c1 ff   9c
37c8 : e0 e7 e7 e7 27 c7 e7 ff   e3
37d0 : 00 9c cf e7 cf 9c 00 ff   f1
37d8 : db ff c3 f9 c1 99 c1 ff   d3
37e0 : db ff c3 99 99 99 c3 ff   54
37e8 : db ff 99 99 99 99 c1 ff   ca
37f0 : c3 99 99 93 99 99 93 9f   4d
37f8 : e7 db ff ff ff ff ff ff   cd
3800 : ff 00 ff 00 ff 00 ff 00   ff
```

Listing 8. Der DIN-Zeichensatz optisch verbessert. Ein Beispiel der Möglichkeiten von Charedit. Bitte beachten Sie die unter »Achtung!« aufgeführten Hinweise.

```
Name : outline (g/k) ": 2000 2802
----------------------------------
2000 : 42 9e 44 ba 44 f2 86 fc  ea
2008 : 00 7c 46 7a c2 bb c5 7f  47
2010 : e0 a0 be 83 bd bd 83 fe  36
2018 : 00 7e c2 be a0 be c2 7e  e8
2020 : 07 05 7d c1 bd bd c1 7f  11
2028 : 00 7e c3 bd 81 bf c2 7e  2e
2030 : 3e 62 de 82 de 50 50 70  3a
2038 : 00 7e c3 bd bd c1 7d 43  86
2040 : e0 a0 be 83 bd a5 a5 e7  00
2048 : 38 28 78 48 68 6c 44 7c  af
2050 : 1c 14 1c 14 14 14 f4 8c  cf
2058 : e0 a0 bc b4 8c b6 ba ee  95
2060 : 78 48 68 28 28 6c 44 7c  0b
2068 : 00 fe 83 ad ad ad bd e7  8d
2070 : 00 fe 83 bd a5 a5 a5 e7  76
2078 : 00 7e c3 bd a5 bd c3 7e  b4
2080 : 00 fe 83 bd bd 83 be a0  cc
2088 : 00 7f c1 bd bd c1 7d 05  5a
2090 : 00 fe 83 bd a7 a0 a0 e0  6c
2098 : 00 7f c1 bf c3 fd 83 fe  f8
20a0 : 38 6e 42 6e 28 2e 32 1e  67
20a8 : 00 e7 a5 a5 a5 bd c1 7f  08
20b0 : 00 e7 a5 a5 bd db 66 3c  8f
20b8 : 00 ee aa ba aa aa d6 7c  86
20c0 : 00 ee ba d6 6c d6 ba ee  07
20c8 : 00 e7 a5 a5 bd c1 7b 46  3e
20d0 : 00 ff 81 f3 66 cf 81 ff  99
20d8 : 7c 44 5c 50 50 5c 44 7c  89
20e0 : 60 50 28 14 0a 05 03 00  ca
20e8 : 7c 44 74 14 14 74 44 7c  15
20f0 : 10 28 44 aa ee 28 28 38  bc
20f8 : 00 00 00 00 00 ff 81 ff  fe
2100 : 00 00 00 00 00 00 00 00  01
2108 : 70 50 50 50 50 70 50 70  69
2110 : 7c 54 54 7c 00 00 00 00  5b
2118 : 7e db 81 db db 81 db 7e  96
2120 : 38 6e c2 ae 82 ea 86 ec  89
2128 : e6 aa f2 24 48 9e aa ce  66
2130 : 70 d8 a8 de aa b6 ca 7e  9b
2138 : 38 38 28 38 00 00 00 00  9e
2140 : 1c 24 48 50 50 48 24 1c  9b
2148 : 70 48 24 14 14 24 48 70  cc
2150 : fe aa c6 92 c6 aa fe 00  65
2158 : 38 28 ee 82 ee 28 38 00  c1
2160 : 00 00 00 00 38 28 58 60  47
2168 : 00 00 fc 84 fc 00 00 00  08
2170 : 00 00 00 00 00 70 50 70  16
2178 : 06 0a 14 28 50 a0 c0 00  9b
2180 : 7c c6 b2 aa 9a ba c6 7c  f5
2188 : 38 68 48 68 28 6c 44 7c  03
2190 : fc 86 fa 1a 24 5e 82 fe  0e
2198 : fc 86 fa 24 3a fa 86 fc  aa
21a0 : 0c 14 24 54 b6 82 f6 1c  de
21a8 : fe 82 be 84 fa ba c6 7c  c1
21b0 : 1c 24 5c 86 ba ba c6 7c  5c
21b8 : fe 82 fa 14 28 28 28 38  0d
21c0 : 7c c6 ba 44 ba ba c6 7c  6c
21c8 : 7c c6 ba ba 82 74 88 f0  7d
21d0 : 70 50 70 00 70 50 70 00  d0
21d8 : 70 50 70 00 70 50 b0 e0  9b
21e0 : 18 28 50 a0 a0 50 28 18  92
21e8 : 00 fc 84 fc 84 fc 00 00  57
21f0 : 60 50 28 14 14 28 50 60  8a
21f8 : 7c c6 ba fa 16 1c 14 1c  b0
2200 : 30 48 24 18 00 00 00 00  61
2208 : 7c c6 ba ba 82 ba aa ee  74
2210 : fc 86 ba 84 ba ba 86 fc  24
2218 : 7c c6 ba ae ae ba c6 7c  51
2220 : f8 8c b6 aa aa b6 8c f8  e6
2228 : 7e c2 be 88 b8 be c2 7e  51
2230 : 7e c2 be 88 b8 a0 a0 e0  a5
2238 : 7c c6 be a0 ae ba c6 7c  b0
2240 : ee aa ba 82 ba aa aa ee  0c
2248 : fe 82 ee 28 28 ee 82 fe  4a
2250 : 0e 0a 0a 0a ea ba c6 7c  c0
2258 : e6 aa b4 88 88 b4 aa ee  88
2260 : e0 a0 a0 a0 a0 be 82 fe  d4
2268 : fe 82 aa aa ba aa aa ee  31
2270 : ee ba 9a aa b2 aa aa ee  c0
2278 : 7c c6 ba aa aa ba c6 7c  f0
2280 : fc 86 ba ba 86 bc a0 e0  58
2288 : 7c c6 92 aa ba 92 c9 7f  c8
2290 : fc 86 ba ba 86 b4 ba ee  ac
2298 : 7e c2 be c4 7a fa 86 fc  53
22a0 : fe 82 ee 28 28 28 28 38  75
22a8 : ee aa aa aa aa ba c6 7c  80
22b0 : ee aa aa aa aa 54 28 10  01
22b8 : ee aa aa ba aa aa 82 fe  05
22c0 : c6 aa 54 28 28 54 aa c6  53
22c8 : ee aa aa 54 28 28 28 38  15
22d0 : fe 82 fa 14 28 5e 82 fe  ce
22d8 : ee 7c c6 ba 82 ba aa ee  94
22e0 : ee 7c c6 ba aa ba c6 7c  aa
22e8 : ee aa ee aa aa ba c6 7c  d1
22f0 : 00 03 fd 83 d4 54 7c 00  43
22f8 : 00 00 00 00 00 ff 81 ff  fe
2300 : ff 81 81 81 81 81 81 ff  7a
2308 : 24 24 24 24 24 24 24 24  08
2310 : 24 24 27 20 20 3f 00 00  10
2318 : 24 24 e7 00 00 ff 00 00  48
2320 : 24 24 e4 04 04 fc 00 00  38
2328 : 24 24 27 20 20 27 24 24  40
2330 : 00 00 ff 00 00 ff 00 00  30
2338 : 24 24 e4 04 04 e4 24 24  68
2340 : 00 00 3f 20 20 27 24 24  28
2348 : 00 00 ff 00 00 e7 24 24  60
2350 : 00 00 fc 04 04 e4 24 24  50
2358 : 24 24 e7 00 00 e7 24 24  a0
2360 : 7e 81 a5 81 a5 99 81 7e  62
2368 : 1c 26 48 84 5c 86 fc 00  28
2370 : 7e 81 a5 81 99 a5 81 7e  12
2378 : 7e 81 a5 81 81 bd 81 7e  59
2380 : 7e 81 bd a1 a1 bf 81 7e  7d
2388 : 3c 42 9d a1 a1 9d 42 3c  09
2390 : 3c 42 b9 a5 b9 a5 42 3c  5b
2398 : 10 28 44 aa ee 28 28 38  64
23a0 : 18 14 fa 81 fa 14 18 00  62
23a8 : 1c 14 14 77 55 22 14 08  89
23b0 : 00 18 28 5f 81 5f 28 18  96
23b8 : 1f 70 c0 87 87 c0 70 1f  af
23c0 : 80 ff 86 fc fc 86 ff 80  86
23c8 : 3f 20 27 24 e4 84 64 3c  e2
23d0 : ff 81 4d 27 27 4d 81 ff  ab
23d8 : e7 7c 46 7a c2 bb c5 7f  fe
23e0 : e7 a5 7e c3 bd bd c3 7e  88
23e8 : ee aa ee aa aa bb c5 7b  d3
23f0 : 7c 82 92 84 82 92 86 9c  f3
23f8 : 3c 42 99 f7 00 00 00 00  bb
2400 : bd 61 bb 45 bb 0d [illegible] [illegible]  15
2408 : ff 83 b9 85 3d 44 3a 80  c8
2410 : 1f 5f 41 7c 42 42 7c 01  e9
2418 : ff 81 3d 41 5f 41 3d 81  47
2420 : f8 fa 82 3e 42 42 3e 80  2e
2428 : ff 81 3c 42 7e 40 3d 81  21
2430 : c1 9d 21 7d 21 af af 8f  25
2438 : ff 81 3c 42 42 3e 82 bc  e9
2440 : 1f 5f 41 7c 42 5a 5a 18  7f
2448 : c7 d7 87 b7 97 93 bb 83  e0
2450 : e3 eb e3 eb eb eb 0b 73  d0
2458 : 1f 5f 43 4b 73 49 45 11  1a
2460 : 87 b7 97 d7 d7 93 bb 83  b4
2468 : ff 01 7c 52 52 52 42 18  42
2470 : ff 01 7c 42 5a 5a 5a 18  69
2478 : ff 81 3c 42 5a 42 3c 81  3b
2480 : ff 01 7c 42 42 7c 41 5f  33
2488 : ff 80 3e 42 42 3e 82 fa  b5
2490 : ff 01 7c 42 58 5f 5f 1f  b3
2498 : ff 80 3e 40 3c 02 7c 01  37
24a0 : c7 91 bd 91 d7 d1 cd e1  d8
24a8 : ff 18 5a 5a 5a 42 3e 80  47
24b0 : ff 18 5a 5a 42 24 99 c3  d0
24b8 : ff 11 55 45 55 55 29 83  ea
24c0 : ff 11 45 29 93 29 45 11  78
24c8 : ff 18 5a 5a 42 3e 84 b9  51
24d0 : ff 00 7e 0c 99 30 7e 00  06
24d8 : 83 bb a3 af af a3 bb 83  26
24e0 : 9f af d7 eb f5 fa fc ff  f5
24e8 : 83 bb 8b eb eb 8b bb 83  ba
24f0 : ef d7 bb 55 11 d7 d7 c7  23
24f8 : ff ff ff ff ff 00 7e 00  f1
2500 : ff ff ff ff ff ff ff ff  ff
2508 : 8f af af af af 8f af 8f  a6
2510 : 83 ab ab 83 ff ff ff ff  c4
2518 : 81 24 7e 24 24 7e 24 81  99
2520 : c7 91 3d 51 7d 15 79 13  b6
2528 : 19 55 0d db b7 61 55 31  e9
2530 : 8f 27 57 21 55 49 35 81  c4
2538 : c7 c7 d7 c7 ff ff ff ff  d2
2540 : e3 db b7 af af b7 db e3  e4
2548 : 8f b7 db eb eb db b7 8f  c3
2550 : 01 55 39 6d 39 55 01 ff  3a
2558 : c7 d7 11 7d 11 d7 c7 ff  ee
2560 : ff ff ff ff c7 d7 a7 9f  78
2568 : ff ff 03 7b 03 ff ff ff  c7
2570 : ff ff ff ff ff 8f af 8f  c9
2578 : f9 f5 eb d7 af 5f 3f ff  55
2580 : 83 39 4d 55 65 45 39 83  0a
2588 : c7 97 b7 97 d7 93 bb 83  0c
2590 : 03 79 05 e5 db a1 7d 01  11
2598 : 03 79 05 db c5 05 79 03  85
25a0 : f3 eb db ab 49 7d 09 e3  62
25a8 : 01 7d 41 7b 05 45 39 83  8e
25b0 : e3 db a3 79 45 45 39 83  03
25b8 : 01 7d 05 eb d7 d7 d7 c7  62
25c0 : 83 39 45 bb 45 45 39 83  13
25c8 : 83 39 45 45 7d 8b 77 0f  12
25d0 : 8f af 8f ff 8f af 8f ff  cf
25d8 : 8f af 8f ff 8f af 4f 1f  15
25e0 : e7 d7 af 5f 5f af d7 e7  2d
25e8 : ff 03 7b 03 7b 03 ff ff  78
25f0 : 9f af d7 eb eb d7 af 9f  56
25f8 : 83 39 45 05 e9 e3 eb e3  3f
2600 : cf b7 db e7 ff ff ff ff  9f
2608 : 83 39 45 45 7d 45 55 11  9b
2610 : 03 79 45 7b 45 45 79 03  fb
2618 : 83 39 45 51 51 45 39 83  df
2620 : 07 73 49 55 55 49 73 07  59
2628 : 81 3d 41 77 47 41 3d 81  fe
2630 : 81 3d 41 77 47 5f 5f 1f  ba
2638 : 83 39 41 5f 51 45 39 83  bf
2640 : 11 55 45 7d 45 55 55 11  73
2648 : 01 7d 11 d7 d7 11 7d 01  45
2650 : f1 f5 f5 f5 15 45 39 83  df
2658 : 19 55 4b 77 77 4b 55 11  27
2660 : 1f 5f 5f 5f 5f 41 7d 01  eb
2668 : 01 7d 55 55 45 55 55 11  9e
2670 : 11 45 65 55 4d 55 55 11  1f
2678 : 83 39 45 55 55 45 39 83  ff
2680 : 03 79 45 45 79 43 5f 1f  a7
2688 : 83 39 6d 55 45 6d 36 80  47
2690 : 03 79 45 45 79 4b 45 11  73
2698 : 81 3d 41 3b 85 05 79 03  dc
26a0 : 01 7d 11 d7 d7 d7 d7 c7  ca
26a8 : 11 55 55 55 55 45 39 83  cf
26b0 : 11 55 55 55 55 ab d7 ef  5e
26b8 : 11 55 55 45 55 55 7d 01  6a
26c0 : 39 55 ab d7 d7 ab 55 39  2c
26c8 : 11 55 55 ab d7 d7 d7 c7  7a
26d0 : 01 7d 05 eb d7 a1 7d 01  d1
26d8 : 11 83 39 45 7d 45 55 11  1b
26e0 : 11 83 39 45 55 45 39 83  15
26e8 : 11 55 11 55 55 45 39 83  fe
26f0 : ff fc 02 7c 2b ab 83 ff  9c
26f8 : ff ff ff ff ff 00 7e 00  f1
2700 : 00 7e 7e 7e 7e 7e 7e 00  85
2708 : db db db db db db db db  07
2710 : db db d8 df df c0 ff ff  0f
2718 : db db 18 ff ff 00 ff ff  e7
2720 : db db 1b fb fb 03 ff ff  07
2728 : db db d8 df df d8 db db  0f
2730 : ff ff 00 ff ff 00 ff ff  2f
2738 : db db 1b fb fb 1b db db  07
2740 : ff ff c0 df df d8 db db  57
2748 : ff ff 00 ff ff 18 db db  2f
2750 : ff ff 03 fb fb 1b db db  4f
2758 : db db 18 ff ff 18 db db  0f
2760 : 81 7e 5a 7e 5a 66 7e 81  5d
2768 : e3 d9 b7 7b a3 79 03 ff  a7
2770 : 81 7e 5a 7e 66 5a 7e 81  cd
2778 : 81 7e 5a 7e 7e 42 7e 81  96
2780 : 81 7e 42 5e 5e 40 7e 81  82
2788 : c3 bd 62 5e 5e 62 bd c3  06
2790 : c3 bd 46 5a 46 5a bd c3  c4
2798 : ef d7 bb 55 11 d7 d7 c7  cb
27a0 : e7 eb 05 7e 05 eb e7 ff  dd
27a8 : e3 eb eb 88 aa dd eb f7  c6
27b0 : ff e7 d7 a0 7e a0 d7 e7  c9
27b8 : e0 8f 3f 78 78 3f 8f e0  c0
27c0 : 7f 00 79 03 03 79 00 7f  f9
27c8 : c0 df d8 db 1b 7b 9b c3  ad
27d0 : 00 7e b2 d8 d8 b2 7e 00  f4
27d8 : 18 83 b9 85 3d 44 3a 80  b1
27e0 : 18 5a 81 3c 42 42 3c 81  37
27e8 : 11 55 11 55 55 44 3a 84  fc
27f0 : 83 7d 6d 7b 7d 6d 79 63  ed
27f8 : c3 bd 66 08 ff ff ff ff  34
2800 : ff ff 00 00 00 00 00 00  ff
```

Listing 9. Umrandeter Zeichensatz als grafisch interessanter Effekt. Bitte beachten Sie die unter »Achtung« aufgeführten Hinweise.

Bildschirm mit 4000 Zeichen durch Interlace

Wer hätte das gedacht? Nicht nur der Commodore Amiga kann die Anzahl der Rasterzeilen auf dem Bildschirm mit Hilfe eines »Interlace«-Modus verdoppeln; auch in den Tiefen des VDC-Chips im C 128 schlummert eine solche Möglichkeit.

Auf dem Commodore Amiga kennt man ihn schon lange: den »Interlace«-Modus, der durch eine trickreiche Schaltung innerhalb des Amiga-Videochips die vertikale Auflösung auf dem angeschlossenen Bildschirm verdoppelt. Das geht allerdings auf Kosten der Bildqualität, da nun wegen interner Zeitprobleme des Videochips das Monitorbild durch zwei gegeneinander versetzte »Halbbilder« aufgebaut wird, was je nach verwendetem Monitor zu einem mehr oder weniger starken Flimmern führt.

Bild 1. Der Bildschirm im Interlace-Modus

Auch der VDC-Chip, der ja im C 128 für die 80-Zeichen-Darstellung zuständig ist, hat das Interlace-Verfahren bereits eingebaut. Um es zu aktivieren, sind lediglich sieben (!) Basic-Zeilen erforderlich, die einige Register des Grafikchips verändern und somit den Interlace-Modus einschalten. Das nun vorhandene Bildschirmflimmern wird allerdings durch eine neue, bisher nicht einmal vom Amiga bekannte Eigenschaft des VDC wieder wettgemacht: Er ist nun fähig, 50 Zeilen (und damit 4000 Zeichen) Text auf dem Bildschirm darzustellen! (Bild 1). Der Interlace-Modus sorgt dafür, daß der Text trotz der nun relativ kleinen Zeichen noch lesbar bleibt.

50 Bildschirmzeilen für den C 128

Das Programm »Interlace« (Listing 1) nimmt die Umschaltung des C 128 auf den Interlace-Modus mit 50 Zeilen auf dem Bildschirm vor. Wir haben es ohne REM-Anweisungen abgedruckt und diese statt dessen getrennt in Listing 2 untergebracht, um Ihnen das Abtippen mit dem neuen Checksummer 128 zu erleichtern. Die REM-Anweisung und die zugehörige Programmzeile haben dabei immer die gleiche Zeilennummer und stehen nebeneinander.

Im Programm erfolgt die eigentliche Umschaltung des VDC in den Zeilen 10 bis 70. Die restlichen Programmzeilen dienen lediglich dazu, den neu gewonnenen Raum auf dem Bildschirm zu beschreiben. Da der eingebaute Bildschirmeditor des C 128 nämlich nur auf die Benutzung der (nun) oberen 25 Zeilen des Bildschirms angelegt ist, müssen die unteren 25 Zeilen durch direktes Verändern des VDC-Speichers beschrieben werden. Im Programm werden als Beispiel einfach die oberen 25 Bildschirmzeilen in die unteren 25 umkopiert. Dieser Vorgang kann im Direktmodus jederzeit mit GOTO 100 eingeleitet werden.

Den VDC richtig ansprechen

Um die mit »Interlace« gebotenen Möglichkeiten auch ausnutzen zu können, ist die Kenntnis der Bedeutung einiger VDC-Register nötig. Nach den Befehlen

```
FAST:BANK 15
A = DEC ("CDCC")
```

```
10 FAST:BANK 15                                          <LBC>
20 A=DEC("CDCC")                                         <CNB>
30 SYSA,77,4                                             <CIK>
40 SYSA, 3,8                                             <2HU>
50 SYSA,0,36:SYSA,64,7                                   <TFU>
60 SYSA,50,6                                             <FQJ>
70 POKE2607,16:SYSA,16,20                                <LDS>
80 PRINT "{CLR}"                                         <3PM>
90 :                                                     <NQV>
100 FOR I=0TO1999STEP200                                 <K3B>
110 SYSA,128,24                                          <IUI>
115 :                                                    <3KE>
120 SYSA,(2000+I)AND255,19                               <59F>
130 SYSA,(2000+I)/256,18                                 <4SK>
140 SYSA,IAND255,33:SYSA,I/256,32                        <QH3>
150 :                                                    <0S3>
160 SYSA,200,30                                          <S6H>
170 SYSA,(6096+I)AND255,19:SYSA,(6096+I)/256,18          <00N>
180 SYSA,(4096+I)AND255,33:SYSA,(4096+I)/255,32          <8VC>
190 SYSA,200,30                                          <26H>
200 NEXT                                                 <180>
```

Listing 1. »Interlace« zum Aktivieren des Interlace-Modus. Bitte mit dem Checksummer 128 eingeben.

```
10 :
20 REM EINSPRUNG DER ROUTINE ZUM SETZEN DER VDC-REGISTER
30 REM VERTIKALE SYNCHRONISATION MIT DOPPELTER GESCHWINDIGKEIT
40 REM INTERLACE-MODUS EINSCHALTEN
50 REM STELLUNG DES MONITORBILDES JUSTIEREN
60 REM EINSTELLUNG DER BILDSCHIRMZEILEN AUF 50
70 REM ATTRIBUTSPEICHER VERSCHIEBEN
80 :
90 REM ROUTINE ZUM UMKOPIEREN DER OBEREN 25 ZEILEN:
100 REM BLOCKVERSCHIEBUNG VON JEWEILS 200 BYTES
110 REM REGISTER 24 MUSS 128 ENTHALTEN
115 REM VERSCHIEBEN DES BILDSCHIRMSPEICHERS
120 REM LOWBYTE DER ZIELADRESSE
130 REM HIGHBYTE DER ZIELADRESSE
140 REM STARTADRESSE DES BLOCKS
150 REM VERSCHIEBEN DES ATTRIBUTSPEICHERS
160 REM 200 BYTES VERSCHIEBEN UND START DES BLOCKVERSCHIEBENS
170 REM ZIELADRESSE
180 REM STARTADRESSE
190 REM 200 BYTES VERSCHIEBEN UND START DES BLOCKVERSCHIEBENS
200 :
```

Listing 2. Die Kommentarzeilen zu »Interlace«. Wegen der besseren Übersicht sind sie nicht in Listing 1 integriert.

Register	Belegung
32/33	Startadresse des Blocks HI/LO
18/19	Zieladresse des Blocks HI/LO
24	muß auf 128 gesetzt werden
30	Anzahl der zu verschiebenden Bytes

Tabelle 1. Die Registerbelegung bei der VDC-internen Blockverschiebung

kann man das VDC-Register RE mit dem Wert WE beschreiben, indem man den Befehl

```
SYS A,WE,RE
```

eingibt.

Da der interne Speicher des VDC, in dem sich auch der Bildschirmspeicher befindet, nicht direkt von außen ansprechbar ist, muß das über verschiedene Register geschehen. Dabei ist zu beachten, daß der VDC eine Speicheradresse in seinen Registern in der Reihenfolge High-Byte/Low-Byte ablegt, was eine gewisse Umgewöhnung erfordert. So beschreibt man eine Speicherstelle des VDC (nur im Programm-Modus anwendbar):

```
SYS A, Low-Byte, 19
SYS A, High-Byte, 18
SYS A, Wert, 31
```

Der VDC verschiebt Blöcke

Im Programm »Interlace« wird die interne Blockverschieberoutine des VDC verwendet, um die oberen 25 Zeilen des Bildschirms in die unteren 25 Zeilen umzukopieren. Da allerdings nur maximal 256 Zeichen auf einmal verschoben werden können, wird die Routine vom Programm mehrmals aufgerufen.

Bei der Verwendung der Blockverschieberoutine muß aber die von »Interlace« geänderte VDC-Speicherkonfiguration (Bild 2) beachtet werden: Da Bildschirm- und Attributspeicher nun den doppelten Speicherplatz wie im Normalmodus benötigen, sind sie im VDC-Speicher teilweise nicht an den »normalen« Stellen zu finden (siehe Bild 2). Die Registerbelegung für die Blockverschiebung steht in Tabelle 1. Das Register 30 muß als letztes beschrieben werden, da durch den Schreibvorgang die Blockverschiebung gestartet wird.

Aus Bild 2 läßt sich noch etwas Interessantes entnehmen: Da für Bildschirm- und Attributspeicher im VDC jeweils 4096 Zeichen zur Verfügung stehen, von »Interlace« aber nur 4000 (50 Zeilen mal 80 Zeichen) genutzt werden, wäre es zusätzlich möglich, mit SYS A,51,6 einen Bildschirm mit 51 Zeilen zu definieren und die 51. Zeile auch noch für Text zu nutzen.

Interlace auch in Maschinensprache

Gerade die Programmierung des VDC über Register ist in Maschinensprache sehr einfach durchführbar. Das folgende Unterprogramm (Assembler-Routine unter Bild 2)

```
setreg   stx $d600
m2       bit $d600
         bpl m2
         sta $d601
         rts
```

setzt das VDC-Register, dessen Nummer im X-Register des Prozessors steht, auf den Wert im Akkumulator. Mit Hilfe dieses Unterprogramms können alle vorher beschriebenen VDC-Eigenschaften auch in Maschinensprache genutzt werden.

Natürlich kann »Interlace« nur eine Anregung für Programmierer sein, sich selbst mit dieser erstaunlichen, bisher unentdeckten Fähigkeit ihres C128 zu beschäftigen und eigene Programme zur Ausnutzung dieses Modus zu entwickeln. Es wäre zum Beispiel denkbar, die unteren 25 Bildschirmzeilen auch für den Cursor zugänglich zu machen.

(Richard Moeskops/Andreas Lietz)

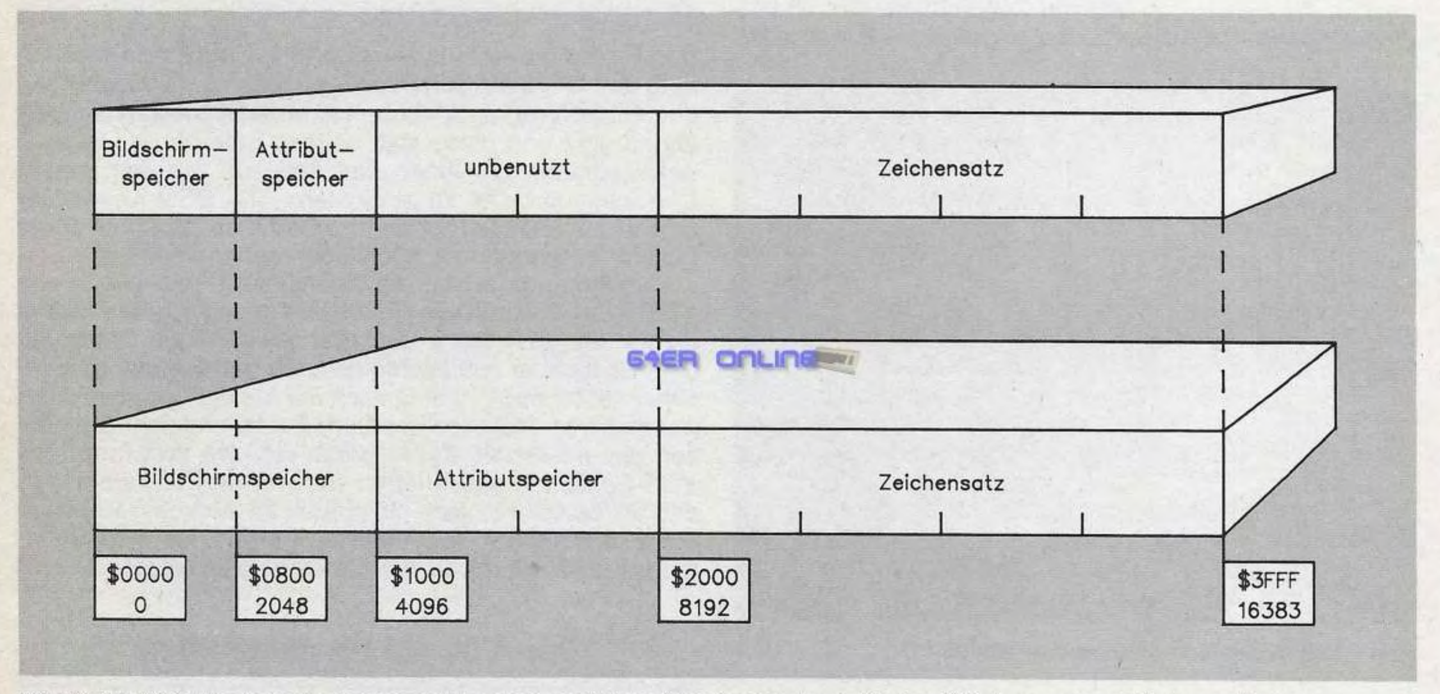


Bild 2. Die VDC-Speicherbelegung mit (unten) und ohne Interlace (oben). Einige Blöcke werden dabei verschoben.

Literatur:
Gerd Möllmann: C 128 Programmieren in Maschinensprache, Markt & Technik Verlag, 269 Seiten, ISBN 3-89090-213-8, 52 Mark
Kurs: »VDC – 80 Zeichen ist nicht alles« in den 64er-Ausgaben 3/87, 5/87, 7/87 und 8/87

2 MHz IRQ: Programme stufenlos beschleunigt

Auch im C64-Modus des C128 läßt sich bekanntlich der Betrieb mit 2 MHz aktivieren. Dabei war bisher allerdings keine vernünftige Bilddarstellung mehr möglich. Wir stellen Ihnen ein Programm vor, das einen Kompromiß aus Geschwindigkeit und guter Optik liefert.

Wie läßt sich nun die hohe Geschwindigkeit mit einer funktionierenden Bilddarstellung in Einklang bringen? Nun, der C64 verfügt bekanntlich über die Möglichkeit des Rasterinterrupts. Damit läßt sich der Bildschirm in unabhängige Fenster aufteilen, in denen der Videochip vollkommen verschiedene Werte verarbeitet. So kann man beispielsweise eine Grafik mit Text am unteren Bildschirmrand darstellen, ein beliebter Trick bei Adventure-Spielen.

Diese Eigenschaft macht sich unser Programm zunutze. Der Bildschirm wird in einen Bereich mit 2-MHz-Prozessortakt und in einen mit 1-MHz-Takt unterteilt. Dazu wird im entsprechenden Bereich das Register in Speicherstelle 53296 ($D030) ein- (Bit 0=1) bzw. ausgeschaltet (Bit 0=0).

Je größer das Fenster mit 2-MHz-Taktfrequenz – also das Fenster mit der chaotischen Bilddarstellung – ist, desto schneller läuft der C64-Modus. Die Obergrenze sind hier natürlich 2 MHz. Je kleiner dieses Fenster ist, desto langsamer arbeitet der Computer. Die Untergrenze stellt dabei 1 MHz dar.

Geben Sie bitte Listing 1 mit dem Checksummer 128 ein. Beachten Sie dazu die Eingabehinweise auf Seite 158. Nach dem Start mit »RUN« kann die Startadresse, ab der das Programm generiert werden soll (üblicherweise 49152)

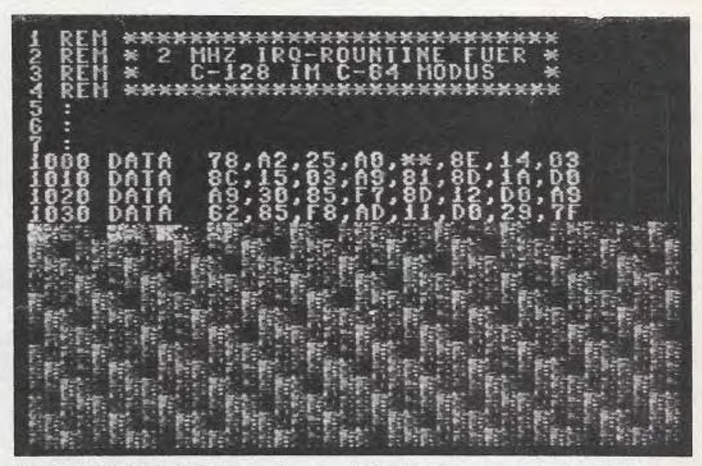


Bild 1. »2 MHz IRQ« nach dem Start. Die chaotische Bilddarstellung im unteren Bereich ist auf den eingeschalteten 2-MHz-Takt zurückzuführen.

eingegeben werden. Diese Adresse wird dabei auf ein Vielfaches von 256 gerundet. Zusätzlich können Sie noch wählen, ob das Programm gespeichert werden soll, oder nicht. Anschließend gibt das Programm aus, mit welchem SYS-Befehl »2 MHz IRQ« aktiviert beziehungsweise abgeschaltet werden kann.

Das vom Lader generierte Programm-File läßt sich nach dem Laden mit angehängtem »,8,1« (also absolut laden) durch »SYS« und der jeweiligen Startadresse starten.

Nach dem Start zeigt sich in der unteren Bildschirmhälfte ein Kräuseln (Bild 1); in diesem Bildschirmteil läuft der Prozessor mit 2-MHz-Takt. Mit den Funktionstasten kann man nun Größe und Position des Bildschirmfensters verändern:

```
1 REM ***************************          <640>
2 REM * 2 MHZ IRQ-ROUNTINE FUER *          <6BP>
3 REM *   C-128 IM C-64 MODUS   *          <6CP>
4 REM ***************************          <6GQ>
5 :                                        <A18>
6 :                                        <APB>
7 :                                        <AHA>
1000 DATA  78,A2,25,A0,**,8E,14,03         <JML>
1010 DATA  8C,15,03,A9,81,8D,1A,D0         <A6A>
1020 DATA  A9,30,85,F7,8D,12,D0,A9         <AD0>
1030 DATA  62,85,F8,AD,11,D0,29,7F         <QHK>
1040 DATA  8D,11,D0,58,60,AD,19,D0         <RTF>
1050 DATA  8D,19,D0,10,34,AD,12,D0         <LCJ>
1060 DATA  C5,F7,D0,18,A9,00,8D,30         <QC4>
1070 DATA  D0,AD,11,D0,29,7F,8D,11         <BBH>
1080 DATA  D0,A5,F7,18,65,F8,8D,12         <35F>
1090 DATA  D0,4C,81,EA,A9,01,8D,30         <KKE>
1100 DATA  D0,AD,11,D0,29,7F,8D,11         <1BH>
1110 DATA  D0,A5,F7,8D,12,D0,4C,81         <MUD>
1120 DATA  EA,A5,CB,C6,F9,D0,53,A0         <MM2>
1130 DATA  02,84,F9,C9,04,F0,3B,C9         <6MQ>
1140 DATA  05,F0,29,C9,06,F0,12,C9         <ILG>
1150 DATA  03,D0,0B,A5,F8,C9,09,90         <RBT>
1160 DATA  05,38,E9,08,85,F8,4C,BA         <MBP>
1170 DATA  **,A5,F8,18,65,F7,C9,FA         <IIE>
1180 DATA  B0,07,A5,F8,18,69,08,85         <RDH>
1190 DATA  F8,4C,BA,**,A5,F7,C9,38         <TC2>
1200 DATA  90,05,38,E9,08,85,F7,4C         <32K>
1210 DATA  BA,**,A5,F7,18,65,F8,C9         <PTU>
1220 DATA  F8,B0,07,A5,F7,18,69,08         <K4L>
1230 DATA  85,F7,AD,0D,DC,58,4C,31         <BI4>
1240 DATA  EA,78,A2,31,A0,EA,8C,15         <290>
1250 DATA  03,8E,14,03,A9,00,8D,30         <C93>
1260 DATA  D0,8D,1A,D0,58,60,END           <SEC>
9998 :                                     <UJR>
9999 :                                     <URQ>
10000 INPUT "STARTADRESSE ";SA             <66H>
10010 PRINT "{UP}ABSPEICHERN (J/N)?{4SPACE}": GET
      X$: IF X$="" THEN 10010              <RN9>
10015 PRINT X$                             <VVI>
10020 SA=INT(SA/256)*256                   <RCH>
10030 XX=SA/256                            <K3C>
10040 RESTORE : POKE 53296,1               <M0F>
10050 READ A$                              <K4N>
10055 IF A$="**" THEN A=XX: GOTO 10100     <0NA>
10057 IF A$="END" THEN 10120               <20F>
10060 AH=ASC(LEFT$(A$,1)): AL=ASC(RIGHT$(A$,1))  <DL3>
10070 IF AH>57 THEN AH=AH-7                <2G7>
10080 IF AL>57 THEN AL=AL-7                <VNJ>
10090 A=(AH-48)*16+(AL-48)                 <K9J>
10100 POKE SA+X,A                          <BIC>
10110 X=X+1: GOTO 10050                    <PDU>
10120 POKE 53296,0: PRINT "{CLR,DOWN,2SPACE}EINSC
      HALTEN: SYS"SA                       <F6H>
10130 PRINT "{DOWN,2SPACE}AUSSCHALTEN: SYS"SA+193
      "{3DOWN}"                            <NGC>
10140 IF X$<>"J" THEN END                  <GFE>
10150 SYS 57812"2MHZ-IRQ"+STR$(SA),8,1     <I6B>
10160 POKE 250,0                           <NF2>
10170 POKE 251,SA/256                      <SA9>
10180 POKE 780,250                         <NIH>
10190 POKE 781,215                         <M6V>
10200 POKE 782,SA/256                      <SG9>
10210 SYS 65496                            <OFV>
10220 OPEN 1,8,15: INPUT#1,ER,ER$: CLOSE 1 <2LE>
10230 PRINT ER;ER$"{2DOWN}"                <KQ5>
10240 IF ER=0 THEN END                     <GML>
10250 PRINT "{UP}NOCHMAL VERSUCHEN (J/N)?{4SPACE}
      ": GET X$: IF X$="" THEN 10250       <M0S>
10260 PRINT X$: GOTO 10140                 <JS0>
```

Listing 1. Dieser Basic-Lader generiert individuelle Versionen des »2 MHz«-Programms. Bitte geben Sie das Programm mit dem Checksummer 128 ein.

<F1> verschiebt das Fenster nach unten
<F3> verschiebt das Fenster nach oben
<F5> vergrößert das Fenster
<F7> verkleinert das Fenster

Man kann nun ein Programm nachladen und mit den Funktionstasten die gewünschte Geschwindigkeit einstellen. Mit zunehmender Geschwindigkeit wird die Bilddarstellung zwar immer unübersichtlicher, aber man kann zumindest kontrollieren, ob der Computer noch rechnet oder eventuell abgestürzt ist.

»2 MHz IRQ« benutzt folgende Speicherstellen:

$F7: Rasterzeile, bei der das Textfenster beginnt (Zeile*8+48)
$F8: Größe des Textfensters in Rasterschritten (Größe in Zeilen*8-2)
$F9: Zähler für die Wiederholfrequenz der Funktionstasten

Mit diesen Adresssen können Sie die 2-MHz-Routine auch von einem Programm aus steuern. In Basic erfolgt dies durch Poke-Befehl in die einzelnen Speicherstellen, in Maschinensprache werden die entsprechenden Befehle (LDA, etc.) dazu verwendet.

Mit dieser Anwendung des Rasterinterrupts wird deutlich, daß dieser auch in der praktischen Programmierung unterstützend eingesetzt werden kann.

(Dierk Peithmann/Axel Pretzsch)

MERGE-Routine für Basic-Programme

Haben Sie sich auch schon geärgert, weil Sie die gleiche Routine in mehreren Basic-Programmen eingeben mußten? Diese einfache MERGE-Routine bringt Abhilfe. Jetzt können Sie Programm-Routinen verknüpfen.

Diese MERGE-Routinen (Listing 1) ist ein rein in Basic 7.0 geschriebenes Programm und läßt sich daher leicht nachvollziehen. Die Funktionstasten werden neu belegt und die neue Belegung anfangs am Bildschirm anzeigt (siehe auch Tabelle 1)

Geben Sie Listing 1 mit dem »Checksummer 128« (Seite 158) ein und speichern Sie es unbedingt vor dem ersten Testlauf! Nach dem Starten mit RUN können Sie Programm-Module verknüpfen.

Folgende Schritte sind nötig, um zwei Basic-Programme miteinander zu verknüpfen:

1. Laden des ersten Programms von Diskette durch Drücken von <F3>.

2. Neunumerierung dieses Programms mit <F7>. Danach ist es, beginnend mit Zeilennummer 1, in Einerschritten durchnumeriert, so daß die höchste auftretende Zeilennummer möglichst gering gehalten wird. Mit <F6> wird das Programm gelistet.

3. Schützen des alten Programms durch <F1>. Damit wird sichergestellt, daß ein weiteres Programm beim Laden das bereits im Speicher stehende Programm nicht beschädigt.

4. Laden des anzuhängenden Programm-Moduls mit <F3>.

5. Verketten der beiden eingeladenen Programme mit <F2>. Sollten nur die zwei bisherigen Programme verknüpft werden, kann jetzt ein letztes Mal umnumeriert und anschließend gespeichert werden. Andernfalls sind die Schritte 2 bis 5 für alle weiteren Programm-Module zu wiederholen.

Alle anzuhängenden Programm-Module müssen natürlich mit einer Zeilennummer beginnen, die größer ist als die höchste Zeilennummer des bereits im Speicher befindlichen Programms. Gegebenenfalls ist eine vorherige Umnumerierung mittels »RENUMBER« vorzunehmen. Eifrigen Programmierern ist deshalb zu empfehlen, sich schrittweise eine Modulsammlung anzulegen, in der jedes Modul mit einer sehr hohen Zeilennummer – beispielsweise 50000 – beginnt.

```
100 PRINT CHR$(14)                                    <IT8>
110 PRINT "{CLR,2DOWN,2SPACE,RVSON,9SPACE}MERGE F
    UER C128{12SPACE,DOWN}"                           <BAI>
120 PRINT "{2SPACE}NEUE FUNKTIONSTASTENBELEGUNG:"     <GUT>
130 PRINT "{DOWN,2SPACE}<F1> ALTEN PROGRAMMBEREIC
    H SCHUETZEN"                                      <SQF>
140 PRINT "{DOWN,2SPACE}<F2> VERKETTEN DER PROGRA
    MME"                                              <HCT>
150 PRINT "{DOWN,2SPACE}<F3> DLOAD"                   <K90>
160 PRINT "{DOWN,2SPACE}<F4> DSAVE"                   <3BV>
170 PRINT "{DOWN,2SPACE}<F5> DIRECTORY"               <TUN>
180 PRINT "{DOWN,2SPACE}<F6> LIST"                    <0PQ>
190 PRINT "{DOWN,2SPACE}<F7> RENUMBER 1,1"            <NS5>
200 PRINT "{DOWN,2SPACE}<F8> DELETE"                  <1VE>
210 PRINT "{DOWN,2SPACE}ALTE TASTENBELEGUNG MIT R
    ESET!{UP}"                                        <82D>
220 :                                                 <0C1>
230 KEY 1,"BANK0:A=PEEK(4624)+PEEK(4625)*256-2:PO
    KE2816,PEEK(45):POKE2817,PEEK(46):POKE46,INT(
    A/256):POKE45,A-256*PEEK(46)"+CHR$(13)            <DRS>
240 KEY 2,"POKE45,PEEK(2816):POKE46,PEEK(2817)"+C
    HR$(13)                                           <B57>
250 KEY 3,"DLOAD"+CHR$(34)                            <JMQ>
260 KEY 4,"DSAVE"+CHR$(34)                            <THS>
270 KEY 5,"DIRECTORY"+CHR$(13)                        <QCA>
280 KEY 6,"LIST"+CHR$(13)                             <2K9>
290 KEY 7,"RENUMBER1,1"+CHR$(13)                      <7LG>
300 KEY 8,"DELETE"                                    <HEI>
310 :                                                 <0S3>
320 NEW                                               <40I>
```

Listing 1. »MERGE« bitte mit dem Checksummer 128 (Seite 158) eingeben

Taste	Belegung
F1	alten Programmbereich schützen
F2	Programme verketten
F3	DLOAD"
F4	DSAVE"
F5	DIRECTORY
F6	LIST
F7	RENUMBER1,1
F8	DELETE

Tabelle 1. Die neue Funktionstastenbelegung bei MERGE

Die dem Programm zugrundeliegende Technik wurde übrigens im Buch »Vom C64 zum C128 - Tips & Tricks« beschrieben. (Wolfgang Becker-Adam/Florian Müller)

Literaturhinweis: Florian Müller, Vom C64 zum C128 - Tips & Tricks, Markt & Technik Verlag, 2. Auflage 1987, 290 Seiten inkl. Diskette, ISBN 3-89090-402-5, 49 Mark

FAST-Modus bei sichtbarem 40-Zeichen-Bildschirm

Der FAST-Befehl des Basic 7.0 ist nicht immer verwendbar, da er den 40-Zeichen-Bildschirm ausblendet. Das Programm »C 128-Faster« ermöglicht nun eine Beschleunigung um immerhin 30 Prozent bei sichtbarem Bildschirm.

Der C 128 gehört nicht gerade zu den schnellsten Computern. Er bietet jedoch die Möglichkeit, den Prozessor mit einer Taktfrequenz von 2 MHz (statt 1 MHz im Normalzustand) arbeiten zu lassen. Wenn Sie allerdings keinen 80-Zeichen-Monitor besitzen, müssen Sie normalerweise auf diesen FAST-Modus verzichten, da der 40-Zeichen-Bildschirm dabei abgeschaltet wird. Hier greift nun das Programm »C 128-Faster« ein. Es beschleunigt den C 128 im 40-Zeichen-Modus um zirka 30 Prozent.

Beschleunigung um 30 Prozent

Wie funktioniert nun diese neue Variante der Beschleunigung? Dazu muß angemerkt werden, daß der 2-MHz-Betrieb auch durch POKE 53296,1 aktiviert werden kann. Dadurch wird aber der Bildschirm nicht abgeschaltet, sondern es entsteht ein undefinierbares Flimmern. Um dieses Flimmern nun zu verhindern und eine normale Bilddarstellung zu ermöglichen schaltet »C 128-Faster« erst dann auf 2-MHz-Betrieb, wenn der Rasterstrahl des Monitors die letzte Bildschirmzeile passiert hat. In der Rasterzeile unmittelbar vor der ersten sichtbaren Bildschirmzeile, wird der Prozessor wieder auf 1 MHz gebremst. So bleibt der Bildschirm trotz des teilweisen 2-MHz-Betriebs erhalten.

Bildschirm bleibt voll erhalten

Allerdings ist zu beachten, daß die GRAPHIC-Modi 2 und 4 (bei diesen werden Grafik und Text gemischt) bei aktiviertem »C 128-Faster« nicht benutzt werden können, ebenso wie Operationen am seriellen Bus (Floppy, Drucker, etc.). Für diese muß der »C 128-Faster« kurzzeitig abgeschaltet werden.

Nun zur Bedienung dieses Utility-Programms. Bitte geben Sie Listing 1 (Basic-Lader) mit »Checksummer 128« ein und speichern Sie es vor dem ersten Start. Geladen und gleichzeitig gestartet wird das Programm mit RUN "C 128-FASTER". Daraufhin wird das eigentliche Programm – eine Maschinenroutine – im Speicher generiert. Mit SYS4864 können Sie das Programm aktivieren und mit SYS4968 abstellen.

Interessierte finden auf der Programmservice-Diskette den Assembler-Quelltext im Hypra-Ass-Format unter dem Filenamen »C 128-FASTER.SRC«. Damit können Assembler-Freaks das Programm genau unter die Lupe nehmen.

(Claus Wilke/Florian Müller)

```
10 RESTORE : FOR I= 4864 TO 4986                          <JCE>
20 READ A$: B=B+DEC(A$)                                   <MQV>
30 NEXT                                                   <SRG>
40 IF B<>14367 THEN PRINT "FEHLER IN DATAS": END          <CQ2>
50 RESTORE : FOR I= 4864  TO 4986                         <JSE>
60 READ A$: A=DEC(A$): POKE I,A                           <IF5>
70 NEXT                                                   <CPG>
80 PRINT "{CLR}C128 FASTER AKTIVIERT"                     <6K5>
90 PRINT "{2DOWN}EINSCHALTEN MIT : SYS 4864"              <M9P>
100 PRINT "{DOWN}ABSCHALTEN{2SPACE}MIT : SYS 4968
    "                                                     <J2S>
110 SYS 4864                                              <J34>
120 DATA 78,A9,1F,8D,14,03,A9,13,8D,15,03,AD,11,D
    0,29,7F,8D,11,D0,A9,32                                <44A>
130 DATA 8D,12,D0,A9,81,8D,1A,D0,58,60,A5,D8,F0,0
    4,29,40,F0,20,AD,11,D0                                <TGL>
140 DATA 29,DF,8D,11,D0,A9,73,85,01,AD,16,D0,29,E
    F,8D,16,D0,AD,2C,0A,8D                                <VBV>
150 DATA 18,D0,AD,19,D0,8D,19,D0,AD,12,D0,C9,FA,B
    0,0D,A9,00,8D,30,D0,A9                                <HRV>
160 DATA FA,8D,12,D0,4C,33,FF,A9,01,8D,30,D0,A9,3
    2,8D,12,D0,4C,65,FA,78                                <THB>
170 DATA A9,00,8D,30,D0,A9,65,8D,14,03,A9,FA,8D,1
    5,03,58,60,20                                         <3G9>
```

Listing 1. »C 128-Faster« wird mit »Checksummer 128« (Seite 158) eingegeben und muß vor dem ersten Start gespeichert werden









Booten leichtgemacht

Die Bootroutine des C 128 kann mehr, als die meisten Besitzer dieses Computers vermuten. »Uniboot« macht ihre verborgenen Fähigkeiten jedem Anwender auf einfache Weise zugänglich. Zusätzlich wird der Aufbau des Bootsektors verständlich erklärt.

Nach einem Reset beziehungsweise nach dem Einschalten des C 128 liest die Bootroutine automatisch den Sektor 0 auf Spur 1 der eingelegten Diskette. Soll nun tatsächlich ein Bootvorgang stattfinden, muß eine spezielle Codierung (»CBM« in den ersten drei Bytes des Sektors) vorhanden sein. Ist dies der Fall, werden vom Betriebssystem weitere Blocks oder gegebenenfalls ein Programmfile geladen. Anschließend startet der C 128 ein Maschinenprogramm, das den Rest des Bootsektors belegen kann. Uniboot hilft dem Anwender beim Erstellen solcher Bootsektoren. Dabei wurde großer Wert auf eine hohe Flexibilität gelegt. Ein professioneller Softwareschutz läßt sich ebenso einfach realisieren wie individueller Programmierkomfort. Im ersten Fall schaltet Uniboot die Unterbrechungsfunktionen (Run/Stop, Restore) sowie den Reset ab und startet ein nachgeladenes Programm-File. Im zweiten Fall holt Uniboot eine individuelle Funktionstastenbelegung sowie neue Farbwerte von Disk und aktiviert sie. In Tabelle 1 finden Sie die vielen Möglichkeiten von Uniboot noch einmal übersichtlich zusammengefaßt.

Vielfältige Bausteine

Das »Geheimnis« dieser Vielseitigkeit liegt in der Verwendung einzelner Module, aus denen sich das vollständige Boot-File zusammensetzt. Ausgehend von einem Grundprogramm, das alle verfügbaren Routinen beinhaltet, können Sie nach Belieben einzelne Programmteile ein- bzw. ausschalten oder auch durch eigene Entwicklungen ergänzen, und erhalten so die optimale Lösung für Ihre individuelle Problemstellung.

- Boot-Sektor automatisch schreiben
- Funktionstastenbelegung speichern und beim Booten aktivieren
- Zusätzlichen Datenblock beim Booten laden
- Benutzer-Routine in Bootfile einbauen
- Nach »Booting...« einen beliebigen Text ausgeben
- C64-Modus einschalten
- Beliebige Farben nach <RUN/STOP RESTORE> einstellen
- Nach <RUN/STOP RESTORE> im 80-Zeichen-Modus Fast-Modus einschalten
- <RUN/STOP>, <RUN/STOP RESTORE> und/oder Reset sperren
- Beim Booten bis zu drei POKE-Befehle ausführen

Tabelle 1. Die Leistungen von Uniboot auf einen Blick

Bedienung des Programms

Geben Sie Listing 1 im 128-Modus mit dem Checksummer 128 ein. Zu Beginn kümmert sich Uniboot um die Belegung der Funktionstasten. Da diese erfahrungsgemäß selten geändert werden (man will sich ja nicht ständig an eine neue Belegung gewöhnen), wurde hier auf ein Menü zur Eingabe verzichtet. Setzen Sie daher vor dem Start des Programms in den Zeilen 1230-1320 die gewünschte Belegung direkt in das Listing ein. Bei den zu belegenden Strings gelten folgende Konventionen:

F$(1)-F$(8)	enthält die Belegung der Funktionstasten,
F$(9)	beinhaltet den Text für <SHIFT>+<RUN/STOP>,
F$(10)	den Text für die HELP-Taste.

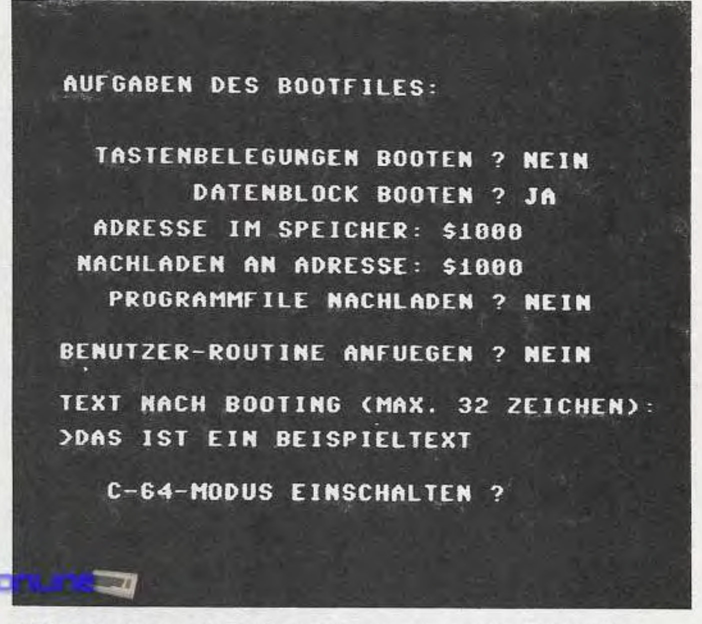


Bild 1. Der erste Teil der Abfragen, um ein individuelles Boot-Pogramm auf der Diskette zu generieren

Der gesamte Text darf dabei nicht länger als 254 Zeichen sein. Die Belegung von <SHIFT>+<RUN/STOP> in Listing 1 ermöglicht übrigens das Laden eines Programms direkt aus dem Directory mit anschließendem Autostart.

Nun sollten Sie »Ihre« Uniboot-Version mit SAVE "Dateiname",8 speichern. Beim anschließenden Start mit RUN zeigt Uniboot die Funktionstastenbelegung an, wie sie auf Wunsch im Bootsektor festgehalten wird.

Funktionstasten belegen

Im folgenden Programmteil stellen Sie sich Ihr eigenes Boot-File zusammen (Bild 1). Wählen Sie mit »J« oder »N« die Aufgaben, die Ihr Boot-File haben soll. Das Programm sortiert anschließend überflüssige Module automatisch aus.

Ein Beispiel soll Ihnen den Einstieg in »Uniboot« etwas erleichtern. In Tabelle 2 finden Sie eine Schritt-für-Schritt-Anleitung, wie der Assembler »Double-Ass« (Seite 66) bootfähig gemacht wird.

Insgesamt stehen folgende Möglichkeiten zur Verfügung (die Reihenfolge entspricht der Programmstruktur):

Tastenbelegung booten

Hier entscheiden Sie, ob die vorher gezeigten Tastenbelegungen beim Booten geladen werden sollen oder nicht. Wenn ja, werden sie auf dem ersten Sektor der ersten Spur der Diskette gespeichert und stehen nach dem Bootvorgang sofort zur Verfügung.

Datenblock booten

Dies ist nur möglich, wenn keine Tastenbelegungen

gebootet werden. Ein ganzer Datenblock (255 Bytes lang) kann mit dieser Funktion an eine beliebige Adresse geladen werden. Geben Sie dazu bei der Abfrage (Bild 1) die Startadresse des Blocks im Speicher (beim Erzeugen des Boot-Files) sowie die Zieladresse beim Booten an. Die Eingabe kann entweder dezimal oder hexadezimal erfolgen, wobei Hex-Adressen durch ein vorangestelltes »$«-Zeichen zu markieren sind.

Der Datenblock kann je nach Inhalt auf unterschiedlichste Weise eingesetzt werden. Man kann damit zum Beispiel Texte direkt in den Bildschirmspeicher laden, ganze Zeropage-Blöcke ändern oder ein kurzes Programm-File bzw. ein Makro nachladen. Hier sind Ihrer Fantasie kaum Grenzen gesetzt.

Individueller Datenblock

Wenn Sie beispielsweise den Block ab $1000 speichern und beim Booten wieder an die gleiche Adresse laden, speichert Uniboot die Funktionstastenbelegung des Computers ab. So können Sie auch ohne Änderung des Programms die Belegung der Funktionstasten sichern. Diese ist in den meisten Fällen nicht mit der Belegung von Uniboot identisch.

Programmfile nachladen

Dieser Punkt ermöglicht das automatische Laden eines Programmfiles. Geben Sie dazu den Filenamen an (Länge maximal 16 Zeichen). Das File wird absolut (also an die Adresse im Programmheader) geladen, daher lassen sich auch Maschinenprogramme problemlos handhaben.

Benutzer-Routine anfügen

Ist im reichhaltigen Angebot der Boot-File-Aufgaben eine von Ihnen benötigte spezielle Problemlösung noch nicht enthalten? Mit dieser Funktion können sie eine eigene, kleine Maschinensprache-Routine an das Grundprogramm des Boot-Files anhängen. Sie muß im Speicher ab $0BB3 laufen und sich während des Ablaufs von »Uniboot« auch an dieser Stelle befinden. Die Gesamtlänge kann 77 Byte betragen. Diese Größe reicht für kleinere Anwendungen völlig aus. Wenn Sie mehr Platz benötigen, geben Sie bei »Datenblock booten« zweimal die Startadresse $0C00 an – damit steht Ihnen der Bereich von $0BB3 bis $0CFF zur Verfügung. Die Routine muß mit RTS enden, da sie als Unterprogramm aufgerufen wird. Durch Aktivieren der Benutzer-Routine verringert sich der Platz für den Boot-Text (folgender Punkt) um 3 Byte.

Eigene Boot-Texte

Text nach Booting

Beim Booten gibt der Computer die Meldung »Booting« aus, der ein beliebiger Text folgen kann. Diesen können Sie hier eingeben. Für den Text sind 16 Zeichen reserviert. Zusätzlich wird der Platz genutzt, den der Filename freiläßt. Maximal sind also 32 Zeichen möglich.

C64-Modus einschalten

Ein Bootfile mit dieser Funktion sollte (außer bei kopiergeschützten Originalprogrammen) auf jeder Ihrer C64-Disketten vorhanden sein. Der Computer springt dann nach dem Booten automatisch in den C64-Modus. Vorher wird allerdings das Programmfile (und der Datenblock) nachgeladen, die dann auch im C64-Modus verfügbar sind. Einzige Bedingung dabei: Die Startadresse des Programm-Files muß größer als $1300 (dezimal 4864) sein. Diese Boot-Variante ist also nicht für C64-Basic-Programme geeignet. Für Assembler oder verschiedene Tools, die

Ein individueller Boot-File für »Double-Ass«:

1. Kopieren Sie »Double-Ass« auf eine neu formatierte Diskette.
2. Belegen Sie mit KEY x,"..." im Direktmodus des C128 die Funktionstasten nach Ihren Wünschen.
2. Laden Sie »Uniboot« mit LOAD "UNIBOOT",8.
3. Nach der Anzeige der Funktionstastenbelegung (die sich höchstwahrscheinlich von Ihrer Belegung unterscheidet) müssen die Abfragen des Programmes so beantwortet werden:

Tastenbelegungen booten	NEIN
Datenblock booten	JA
Adresse im Speicher	$1000
Nachladen an Adresse	$1000
Programmfile nachladen?	JA
Filename	DOUBLE-ASS
Benutzer-Routine anfügen?	NEIN
Text nach Booting:	DOUBLE-ASS
C64-Modus einschalten?	NEIN
Default-Farbwerte ändern?	JA
Reset-Taste inaktivieren?	NEIN
Poke-Befehl ausführen?	NEIN
Autostart durchführen?	JA
Farbwahl: Wählen Sie Ihre »persönlichen Farben«	
80-Zeichen-Werte einstellen?	JA
Folgen Sie den Anweisungen auf dem Bildschirm	
Directory zeigen?	NEIN
Wollen Sie abspeichern?	JA

4. Diskette mit »Double-Ass« einlegen und Boot-File speichern. Sobald der Bootblock auf die Diskette geschrieben wurde, können Sie ihn durch einen Druck auf die Reset-Taste testen.

Tabelle 2. So wird »Double-Ass« (Seite 66) mit einem eigenem Bootsektor versehen

Byte	Inhalt und Funktion
000	CBM-Kennung für Boot-Block
003	Adresse für den Datenblock
005	Nummer der Bank, in die geladen wird
006	Anzahl der nachzuladenden Blöcke
007	Bootmeldung
	$00 am Ende
bis	Filename
	$00 am Ende
041	Bereich, der wahlweise die Daten für
	drei POKEs
	einen POKE und den Zeiger auf die Default-Farben-Routine
	einen POKE und die Sperrung von <RUN/STOP RESTORE>
	zwei POKEs und <RUN/STOP> sperren
	zwei POKEs und <RESTORE> sperren
	enthält, je nachdem welche Konfiguration bei »Default-Farben«, »RUN/STOP sperren« und »RESTORE sperren« gewählt wurde
056	Autostart-Modul
073	Default-Farben-Routine ändert den Warmstart-Vektor und ruft folgendes Unterprogramm auf:
097	Farben-Unterprogramm wird in jedem Fall nach dem Booten aufgerufen, gegebenenfalls auch nach <RESTORE>. Ändert Vorder-/Hintergrund- und Schriftfarbe im 40-Zeichen-Modus bzw. Schriftfarbe und Fast-Modus bei 80-Zeichen-Darstellung.
107	C 64-Reset schreibt in Bank 0 den Text »CBM80« an Adresse $8004 und richtet den Reset-Vektor auf die Killer-Routine
118	C 128-Reset schreibt in Bank 1 an Adresse $FFF8 den Sprungbefehl zur Killer-Routine
152	Killer-Routine wird bei Reset aufgerufen, erzeugt Farbstreifenmuster in einer Endlosschleife
166	Daten der Routinen (CMB80, RUN <CR>)
178	frei für Benutzer-Routine
bis	
255	

Tabelle 3. So belegt »Uniboot« den Bootsektor

sich meistens im RAM-Bereich ab $8000 bzw. $C000 befinden, ist diese Option aber ideal.

Falls der C64-Modus aktiviert wurde, werden die nun folgenden Wahlmöglichkeiten übersprungen, da sie nur im C128-Modus funktionieren.

Default-Farbwerte ändern

Am Ende dieses Menüs können Sie sich Ihre Lieblings-Farbzusammenstellung aussuchen. Dieses Modul sorgt nun dafür, daß nach jedem <RUN/STOP RESTORE> statt den üblichen Farben wieder diese Wunschfarben eingesetzt werden. Außerdem wird im 80-Zeichen-Modus nach <RUN/STOP RESTORE> automatisch der Fast-Modus aktiviert. Sollten Sie die Farbwerte geändert haben, sind natürlich die nächsten beiden Punkte unsinnig und werden daher übersprungen.

Gliederung Zeile	Funktion
1100-1160	Initialisieren
1200-1290	Funktionstastenbelegung
1330-1360	Anzeige der Funktions-Tastenbelegung geschieht einschließlich der Grafikzeichen, was durch Simulation von <INSERT> möglich wird
1350	Aussortieren von <CR> und <SHIFT> <CR>
1380-1390	Zusammenbauen der Texte zu Block 2. Format: Länge der 10 Texte, dann die Texte hintereinander ohne Trennzeichen, wird direkt nach $1000 geladen
1400	Decodieren der Hex-Datas für Block 1
1420-2110	Auswahl der Boot-File-Funktionen
1470 + 1500	Bearbeiten der Eingabe
1520	Einlesen des Datenblocks aus dem Speicher
1530	Sprungbefehl zur Benutzer-Routine einfügen
1570	Benutzer-Routine aus Speicher einlesen
1620-1630	Boot-Text zerlegen und in Block 1 schreiben
1640-1650	Filename zerlegen und in Block 1 schreiben
1670	Sprung in den C 64-Modus einfügen
1710	<RESTORE> ausschalten
1730	<RUN/STOP> ausschalten
1770	Reset sperren
1800-1860	POKE-Befehl durch LDA #xx : STA yyyy erzeugen
1920	Autostart-Modul einschalten
1930-2040	Farbenmenü
2050-2110	Unterprogramm zum Farbenmenü - Tasten abfragen - Werte setzen
2150-2510	Daten speichern
2190	Directory zeigen
2200	Datenkanäle zur Floppy öffnen
2220-2230	Die Datenblöcke probeweise belegen. Tritt ein Fehler auf, sind sie schon belegt.
2240-2270	Sind die ersten 3 Zeichen in Block 1 »CBM«? Nein: Programm-File Ja: Boot-File
2280-2300	Steht ab Byte 175 der Text »RUN«? Ja: Uniboot-File Nein: Fremdes Boot-File
2310-2370	Gegebenenfalls fragen, ob trotz Programm/fremdem Boot-File gespeichert werden soll
2410-2500	Blöcke endgültig belegen
2510	Datenkanäle schließen, Programmende
2520-2540	TRAP-Routine: Bei Fehler Warnton + Close + Help
2550-2570	Ja/Nein-Abfrage
2580-2610	Eingabe der Daten
2650-2760	Daten für Block 1

Tabelle 4. »Uniboot« nach Programmzeilen gegliedert

RESTORE-Taste sperren

Die Kombination <RUN/STOP RESTORE> wird hiermit unwirksam gemacht. Die RUN/STOP-Taste funktioniert aber wie gewohnt.

STOP-Taste sperren

Falls Sie diese Funktion in Ihr Boot-File einbauen, ist eine Unterbrechung durch <RUN/STOP> nach dem Booten nicht mehr möglich. Die Kombination <RUN/STOP RESTORE> funktioniert allerdings nach wie vor. Außerdem verliert die Sperre nach dem ersten Betätigen der RESTORE-Taste Ihre Wirkung. Deshalb ist diese Option nur dann sinnvoll, wenn ein versehentliches Unterbrechen des Programms verhindert, ein gewollter Ausstieg aber weiterhin machbar sein soll.

Reset-Taste inaktivieren

Dies ist für einen guten Programmschutz interessant. Wenn Sie dazu noch die RESTORE-Taste sperren, ist Ihr Programm vor fremden Zugriffen geschützt. Sie können wählen, ob der Reset im C128- oder im C64-Modus gesperrt ist. Nach dem Reset wird ein Farbstreifenmuster erzeugt, ansonsten passiert nichts. Der Rechner ist nur noch mehr durch Aus- und Einschalten wiederzubeleben, wobei natürlich alle Programme gelöscht werden.

Programme schützen

Natürlich können Sie die letzten drei Punkte auch beliebig miteinander kombinieren, um so Ihre Programme optimal zu schützen.

POKE-Befehl ausführen

Eine unkomplizierte, universelle Funktion für alle, die nicht gleich ein komplettes Programm schreiben wollen. Durch die POKEs können Sie einige Sonderwünsche, wie etwa einen List- oder Save-Schutz, problemlos ausführen. Geben Sie die POKE-Adresse und den POKE-Wert an. Es wird dabei immer die Bank 15 beschrieben, alle Zeropage-Adressen sind von dort erreichbar. Mindestens ein POKE-Befehl steht zur Verfügung. Haben Sie die STOP- beziehungsweise RESTORE-Taste nicht gesperrt, erhöht sich die maximale Anzahl um jeweils eins, Sie können also im Höchstfall drei POKEs einsetzen. Hier einige Beispiele für nützliche POKEs:

Listschutz:	POKE 775,139
Tastatur abschalten:	POKE 2592,0
Tasten-Repeat aus:	POKE 2594,0
Save-Schutz:	POKE 818,50

Autostart durchführen

Dieses Modul ist nur wählbar, wenn auch ein Programm-File nachgeladen wird. Es schreibt die Zeichenfolge RUN <RETURN> (CHR$ (13)) in den Tastaturpuffer und führt so nach dem Ladevorgang einen Autostart aus.

Farbe nach Wunsch

Nach dem Zusammenstellen der Modulteile legen Sie im Farbmenü die gewünschte Farbkombination fest. Nur wenn



Liste der wichtigen Variablen	
Felder	
F$(10)	Funktionstastenbelegung
B1(255)	Codes für den Boot-Block
B2(255)	Codes für den Datenblock
Strings	
A$	Enthält die Eingaben
N$	Filename
TX$	Text nach »Booting«
Numerische Variablen	
I,J	Schleifenzähler
L	maximale Länge - der Eingabe - der Funktionstastenstrings - des Boot-Textes
AD	POKE-Adresse bzw. Adresse, an die der Datenblock geladen wird
W	POKE-Wert bzw. Adresse des Datenblocks im Speicher
C,P	Block-Zähler, speichern den Platzbedarf einzelner Module
Z	Cursorzeile
S	Cursorspalte
Flags	
FT = 1	wenn keine Tastenbelegungen und kein Datenblock gespeichert werden sollen
PF = 1	wenn ein Programm-File geladen werden soll
RS = 1	wenn im C128-Modus der Reset gesperrt wurde
BR = 1	wenn <RUN/STOP> gesperrt sein soll

Tabelle 5. Die wichtigsten Variablen von »Uniboot« finden Sie hier übersichtlich aufgeteilt

»Default-Farbwerte ändern« eingestellt wurde, bleiben sie auch nach <RESTORE> erhalten. Die Farben wählen Sie durch Betätigen der Cursortasten. <RETURN> bestätigt die Eingabe.

Wenn Sie mit Ihrer individuellen Kombination fertig sind, werden die Daten auf Diskette gespeichert. Legen Sie dazu die Diskette in Ihr Laufwerk ein, die das Boot-File enthalten soll. Im Zweifelsfall können Sie sich vorher das Directory zeigen lassen.

Das Programm führt vor dem Speichern einen Block-Test durch und teilt Ihnen mit, ob die benötigten Blöcke frei oder von einem alten Boot-File, von einem Programm-File oder von einem fremden Boot-File (zum Beispiel bei CP/M) belegt sind. In den beiden ersten Fällen ist das Speichern problemlos möglich, bei einer Belegung durch ein Programm-File oder einen fremden Boot-Sektor riskieren Sie Datenverlust, wenn Sie trotzdem speichern!

Sobald Sie »grünes Licht« gegeben haben, wird gespeichert. Die Blöcke werden in der BAM als belegt gekennzeichnet, um ein versehentliches Überschreiben zu verhindern. Die Anzahl der freien Blocks beträgt daher nur noch 662.

Zum Ausprobieren Ihres erzeugten Boot-Files drücken Sie einfach die Reset-Taste. Wie gewohnt erscheint zuerst die Einschaltmeldung. Dann allerdings wird die Meldung »Booting«, gefolgt von Ihrem Text, ausgegeben.

Nach dem Laden des Datenblocks läuft die Floppy gegebenenfalls erneut an, um das Programm-File zu laden, danach wird gegebenenfalls die Benutzer-Routine (siehe oben) gestartet. Erst dann springt der Computer in den Direktmodus.

So arbeitet »Uniboot«

Soviel zu den Möglichkeiten, die Sie mit Uniboot haben. Doch wie wurde all dies softwaremäßig verwirklicht?

Die Antwort auf diese Frage soll Ihnen Tabelle 3 in Verbindung mit den Tabellen 4 und 5 geben. Hier finden Sie den Aufbau des Bootsektors von Uniboot, eine Gliederung des Programms und die Liste der wichtigsten Variablen. Betrachten Sie weiterhin Listing 1.

Zum Schluß noch ein Tip: Wollen Sie einen optimalen Software-Schutz realisieren, wird ein kurzes Ladeprogramm notwendig, welches das zu schützende Programm nachlädt. Die Unterbrechungstasten sowie der Reset werden nämlich erst nach dem Laden der Programm-Files gesperrt. (Christian Paul/Axel Pretzsch)

```
1000 REM ************************                 <34C>
1010 REM *  UNIVERSAL-AUTOBOOT  *                 <3CD>
1020 REM *       FUER C128      *                 <3KE>
1030 REM *======================*                 <3SF>
1040 REM *  IDEE & PROGRAMM VON *                 <248>
1050 REM *    CHRISTIAN PAUL    *                 <2C9>
1060 REM *      ROSENWEG 4      *                 <2KA>
1070 REM *   6300 GIESSEN 21    *                 <2SB>
1080 REM ************************                 <144>
1090 :                                            <5GN>
1100 TRAP 2520: IF RWINDOW(2)=40 THEN 1120        <KQJ>
1110 PRINT "BITTE SCHALTEN SIE AUF 40-ZEICHEN UM!
     " CHR$(27)"X"                                <BK1>
1120 COLOR 0,1: COLOR 4,1: SCNCLR : OPEN 3,0      <07N>
1130 PRINT CHR$(19) CHR$(19): PRINT "{RED}*******
     ***********"                                 <1NG>
1140 PRINT "{YELLOW}UNIVERSAL-AUTOBOOT"           <QTM>
1150 PRINT "{RED}*******************{LIG.GREEN}":
     PRINT                                        <Q1R>
1160 DIM F$(10),B1(255),B2(255): BANK 15          <05I>
1170 :                                            <3GF>
1180 REM FUNKTIONSTASTEN-BELEGUNGEN               <1K6>
1190 :                                            <4GJ>
1200 F$(1)=CHR$(7)+"DIRECTORY"+CHR$(13)           <I88>
1210 F$(2)=CHR$(7)+"MONITOR"+CHR$(13)             <KF9>
1220 F$(3)="DLOAD"+CHR$(34)+CHR$(20)+CHR$(34)     <M7N>
1230 F$(4)="DSAVE"+CHR$(34)+CHR$(20)+CHR$(34)     <7M8>
1240 F$(5)="POKE"                                 <6DN>
1250 F$(6)="PRINT PEEK({5SPACE}){6LEFT}"          <DKE>
1260 F$(7)=CHR$(7)+"LIST"+CHR$(13)                <E3C>
1270 F$(8)=CHR$(7)+"RUN"+CHR$(13)                 <RC1>
1280 F$(9)=CHR$(7)+"DLOAD"+CHR$(9)+CHR$(9)+CHR$(9
     )+CHR$(157)+":"+CHR$(13)+CHR$(7)+"RUN:"+CHR$
     (13)                                         <PL4>
1290 F$(10)=CHR$(141)+"UNIVERSAL-AUTOBOOT"+CHR$(1
     41)+" IDEE & PROGRAMM:"+CHR$(141)+"{2SPACE}C
     HRISTIAN PAUL"+CHR$(141)+CHR$(141)           <LA3>
1300 :                                            <209>
1310 REM DATENBLOECKE KONSTRUIEREN                <2SB>
1320 :                                            <30D>
1330 PRINT : PRINT "FUNKTIONSTASTEN-BELEGUNGEN:"
     CHR$(159): PRINT : PRINT                     <H01>
1340 FOR I=1 TO 10: L=LEN(F$(I)): GL=GL+L: B2(I-1
     )=L: PRINT "{WHITE}KEY"I": ";CHR$(159): POKE
      245,L                                       <BJ8>
1350 FOR J=1 TO L: A$=MID$(F$(I),J,1): IF A$=CHR$
     (13) OR A$=CHR$(141) THEN A$="←"             <PN4>
1360 PRINT A$;: NEXT : PRINT : NEXT               <78R>
1370 IF GL>245 THEN PRINT "{2DOWN}ZUVIEL TEXT !":
      GOTO 2510                                   <PMM>
1380 P=10: FOR I=1 TO 10: FOR J=1 TO LEN(F$(I))   <HBG>
1390 B2(P)=ASC(MID$(F$(I),J)): P=P+1: NEXT : NEXT
                                                  <EPG>
1400 FOR I=0 TO 178: READ A$: B1(I)=DEC(A$): NEXT
      : SLOW                                      <34P>
1410 PRINT "{2DOWN,LIG.GREEN,RVSON}TASTE{RVOFF}";
     : GET KEY A$                                 <QBG>
1420 SCNCLR : PRINT "{YELLOW,DOWN}AUFGABEN DES BO
     OTFILES:": PRINT "{RED}YYYYYYYYYYYYYYYYYYYYYY
     YY"                                          <H79>
1430 PRINT "{2DOWN}";CHR$(159);"{2SPACE}TASTENBEL
     EGUNGEN BOOTEN ?";: GOSUB 2550: IF A$="J" TH
     EN 1540                                      <UI8>
1440 PRINT : PRINT "{8SPACE}DATENBLOCK BOOTEN ?";
     : GOSUB 2550: IF A$="J" THEN 1460            <43U>
1450 B1(6)=0: FT=1: GOTO 1530                     <I8I>
1460 PRINT : PRINT "{2SPACE}ADRESSE IM SPEICHER:
     ";: L=5: GOSUB 2580                          <ONK>
1470 IF LEFT$(A$,1)="$" THEN W=DEC(RIGHT$(A$,LEN(
     A$)-1)): ELSE W=VAL(A$)                      <B8H>
1480 IF W>65280 OR W<0 OR W<>INT(W) THEN PRINT "{
     2UP}";: GOTO 1460                            <V0F>
```

Listing 1. »Uniboot« bitte mit dem Checksummer 128 (Seite 158) eingeben

```
1490 PRINT : PRINT " NACHLADEN AN ADRESSE: ";: L=
     5: GOSUB 2580: AD=VAL(A$)                          <G03>
1500 IF LEFT$(A$,1)="$" THEN AD=DEC(RIGHT$(A$,LEN
     (A$)-1)): ELSE AD=VAL(A$)                          <0K3>
1510 IF AD>65280 OR AD<0 OR AD<>INT(AD) THEN PRIN
     T "{2UP}";: GOTO 1490                              <QUA>
1520 FOR I=0 TO 255: B2(I)=PEEK(W+I): NEXT              <7HG>
1530 B1(4)=INT(AD/256): B1(3)=AD-256*B1(4)              <B72>
1540 PRINT "{DOWN,3SPACE}PROGRAMMFILE NACHLADEN ?
     ";: GOSUB 2550: IF A$="N" THEN PF=1: GOTO 15
     60                                                 <7M3>
1550 PRINT "{DOWN}FILENAME: ";: L=16: GOSUB 2580:
      N$=A$                                             <U1I>
1560 PRINT "{2DOWN}BENUTZER-ROUTINE ANFUEGEN ?";:
      GOSUB 2550: IF A$="N" THEN 1600                   <M2T>
1570 EX=3: B1(38)=32: B1(39)=179: B1(40)=11             <T9V>
1580 PRINT : PRINT "STARTADRESSE: $0BB3 = 2995"         <DFE>
1590 FOR I=179 TO 255: B1(I)=PEEK(2816+I): NEXT         <QCQ>
1600 L=32-LEN(N$)-EX: PRINT : PRINT "{DOWN}TEXT N
     ACH BOOTING (MAX."L"ZEICHEN):"                     <TQT>
1610 PRINT : PRINT ">";: GOSUB 2580: TX$=A$             <9LL>
1620 C=7: FOR I=1 TO LEN(TX$): B1(C)=ASC(MID$(TX$
     ,I)): C=C+1: NEXT                                  <JER>
1630 B1(C)=0: C=C+1: IF N$="" THEN 1650                 <HSF>
1640 FOR I=1 TO LEN(N$): B1(C)=ASC(MID$(N$,I)): C
     =C+1: NEXT                                         <GJF>
1650 B1(C)=0: FOR I=C+1 TO 40-EX: B1(I)=234: NEXT
                                                        <OPM>
1660 PRINT : PRINT "{DOWN,3SPACE}C-64-MODUS EINSC
     HALTEN ?";: GOSUB 2550                             <BUP>
1670 IF A$="J" THEN B1(41)=76: B1(42)=75: B1(43)=
     226: GOTO 2150                                     <ELR>
1680 C=10: PRINT "{DOWN}DEFAULT-FARBWERTE AENDERN
      ?";: GOSUB 2550                                   <05L>
1690 IF A$="J" THEN 1740                                <S9G>
1700 C=0: PRINT "{DOWN,4SPACE}RESTORE-TASTE SPERR
     EN ?";: GOSUB 2550: IF A$="N" THEN 1720            <K6E>
1710 B1(41+C)=169: B1(42+C)=98: B1(43+C)=141: B1(
     44+C)=24: B1(45+C)=3: C=5                          <I1J>
1720 PRINT "{DOWN,7SPACE}STOP-TASTE SPERREN ?";:
     GOSUB 2550: IF A$="N" THEN 1740                    <Q23>
1730 B1(41+C)=169: B1(42+C)=112: B1(43+C)=141: B1
     (44+C)=40: B1(45+C)=3: C=C+5                       <R8P>
1740 PRINT "{DOWN} RESET-TASTE INAKTIVIEREN ?";:
     GOSUB 2550: IF A$="N" THEN 1800                    <41A>
1750 PRINT "{DOWN} ... IM C-128-MODUS ?";: GOSUB
     2550: RS=0: IF A$="J" THEN RS=1                    <DLK>
1760 PRINT "{DOWN,2SPACE}... IM C-64-MODUS ?";: G
     OSUB 2550: IF A$="N" THEN 1790                     <KNU>
1770 B1(71)=107: IF RS=0 THEN B1(118)=76: B1(119)
     =79: B1(120)=11                                    <87G>
1780 GOTO 1800                                          <484>
1790 IF RS=1 THEN B1(71)=118                            <MNF>
1800 PRINT "{DOWN,3SPACE}POKE-BEFEHL AUSFUEHREN ?
     ";: GOSUB 2550: IF A$="N" THEN 1880                <565>
1810 PRINT "{DOWN,2SPACE}AN WELCHE ADRESSE POKEN
     ? ";: L=5: GOSUB 2580: AD=VAL(A$)                  <8TT>
1820 IF AD<0 OR AD>65535 OR AD<>INT(AD) OR ASC(A$
     )>57 THEN PRINT "{2UP}";: GOTO 1810                <9R0>
1830 PRINT TAB(17)"{DOWN}POKE-WERT: ";: L=3: GOSU
     B 2580: PW=VAL(A$)                                 <LA9>
1840 IF PW<0 OR PW>255 OR PW<>INT(PW) OR ASC(A$)>
     57 THEN PRINT "{2UP}";: GOTO 1830                  <8I4>
1850 PRINT : B1(41+C)=169: B1(42+C)=PW: B1(43+C)=
     141                                                <LD6>
1860 H=INT(AD/256): B1(44+C)=AD-H*256: B1(45+C)=H
     : C=C+5                                            <3LQ>
1870 IF C<15 THEN 1800                                  <E0T>
1880 IF C>5 THEN 1900                                   <HPF>
1890 FOR I=C TO 10: B1(41+I)=234: NEXT                  <F0K>
1900 IF PF THEN 1920                                    <5DK>
1910 PRINT "{DOWN,3SPACE}AUTOSTART DURCHFUEHREN ?
     ";: GOSUB 2550: IF A$="J" THEN 1930                <563>
1920 B1(56)=76: B1(57)=70: B1(58)=11                    <1MN>
1930 IF RWINDOW(2)=80 THEN PRINT "BITTE 40-ZEICHE
     N-MODUS EINSCHALTEN !!!" CHR$(27)"X                <MSP>
1940 SCNCLR : PRINT "{2DOWN,YELLOW}FARBWAHL 40-ZE
     ICHEN-SCHIRM"                                      <V2B>
1950 PRINT "{RED}YYYYYYYYYYYYYYYYYYYYYYYYYY{2DOWN
     }";CHR$(159)                                       <J95>
1960 PRINT "{5SPACE}RAHMENFARBE:": AD=53280: GOSU
     B 2050: B1(80)=W                                   <UCD>
1970 PRINT "{DOWN}HINTERGRUNDFARBE:": AD=53281: G
     OSUB 2050: B1(85)=W                                <L56>
1980 PRINT "{DOWN,4SPACE}ZEICHENFARBE:": AD=241:
     GOSUB 2050: B1(90)=W                               <909>
1990 PRINT "{2DOWN}80-ZEICHEN-WERTE EINSTELLEN ?"
     ;: GOSUB 2550: IF A$="N" THEN 2150                 <DM6>
2000 IF RWINDOW(2)=40 THEN PRINT "{DOWN}BITTE AUF
      80 ZEICHEN UMSCHALTEN !!!" CHR$(27)"X             <UDE>
2010 SCNCLR : PRINT "FARBWAHL 80-ZEICHEN-SCHIRM":
      PRINT "TTTTTTTTTTTTTTTTTTTTTTTTTT"                <0U3>
2020 PRINT "{2DOWN,4SPACE}ZEICHENFARBE:": AD=241:
      GOSUB 2050: B1(98)=W                              <FJE>
2030 IF RWINDOW(2)=40 THEN 2150                         <M6F>
2040 PRINT "{3DOWN}BITTE WIEDER AUF 40-ZEICHEN UM
     SCHALTEN !!!" CHR$(27)"X": GOTO 2150               <GI7>
2050 W=PEEK(AD) AND 15                                  <OSB>
2060 PRINT "{UP}" TAB(17)W+1"{LEFT} ": GET KEY A$       <49J>
2070 IF A$="{RIGHT}" THEN W=W+1: IF W=16 THEN W=0       <ORJ>
2080 IF A$="{LEFT}" THEN W=W-1: IF W=-1 THEN W=15       <JJH>
2090 IF A$=CHR$(13) THEN RETURN                         <OMH>
2100 POKE AD,W: IF AD=241 THEN PRINT "{UP,4SPACE}
     ZEICHENFARBE:"                                     <UVT>
2110 GOTO 2060                                          <0GU>
2120 :                                                  <70T>
2130 REM DATENBLOECKE ABSPEICHERN                       <0C1>
2140 :                                                  <40H>
2150 BR=1: SCNCLR : PRINT "{YELLOW}DATEN ABSPEICH
     ERN": PRINT "{RED}*****************";CHR$(15
     9)                                                 <REQ>
2160 PRINT : PRINT : PRINT "BITTE LEGEN SIE DIE D
     ISKETTE EIN,"                                      <2M4>
2170 PRINT "DIE DEN AUTOBOOT-FILE ERHALTEN SOLL."
     : PRINT : PRINT                                    <V4U>
2180 PRINT "DIRECTORY ZEIGEN ?";: GOSUB 2550: IF
     A$="N" THEN 2200                                   <B47>
2190 PRINT "{LIG.GREEN}": DIRECTORY : PRINT CHR$(
     159): GOTO 2180                                    <N7B>
2200 FAST : OPEN 1,8,15: OPEN 2,8,13,"#": BR=0          <IF2>
2210 PRINT : PRINT : PRINT "ERGEBNIS DES BLOCK-TE
     STS:{LIG.GREEN}": PRINT                            <A35>
2220 PRINT#1,"B-A 0 1 0": IF DS<>0 THEN 2240            <MPP>
2230 PRINT#1,"B-A 0 1 1": IF DS=0 THEN 2400             <J70>
2240 PRINT#1,"U1 13 0 1 0": N$=""                       <D20>
2250 FOR I=1 TO 3: GET #2,A$: N$=N$+A$: NEXT            <5IH>
2260 IF DS<>0 THEN PRINT DS$: GOTO 2160                 <FH3>
2270 IF N$<>"CBM" THEN 2350                             <VU5>
2280 PRINT#1,"B-P 13 175": N$=""                        <VME>
2290 FOR I=1 TO 3: GET #2,A$: N$=N$+A$: NEXT            <6IT>
2300 IF N$<>"RUN" THEN 2330                             <E7B>
2310 PRINT "DER AUF DIESER DISKETTE BEREITS VOR-"       <73T>
2320 PRINT "HANDENE BOOT-FILE WIRD UEBERSCHRIEBEN
     .": GOTO 2410                                      <PJ3>
2330 PRINT "AUF DIESER DISKETTE BEFINDET SICH"          <LDC>
2340 PRINT "BEREITS EIN FREMDER AUTOBOOT-FILE !":
      GOTO 2370                                         <M3H>
2350 PRINT "DIE BLOECKE SIND BEREITS VON"               <FL7>
2360 PRINT "EINEM PROGRAMMFILE BELEGT !"                <9V2>
2370 PLAY "U9T8Q01C": SLOW : PRINT CHR$(159)"{2DO
     WN}WOLLEN SIE TROTZDEM SPEICHERN ?";               <6BF>
2380 GOSUB 2550: IF A$="N" THEN CLOSE 2: CLOSE 1:
      GOTO 2150                                         <GBG>
2390 FAST : GOTO 2430                                   <9C1>
2400 PRINT "EIN NEUER AUTOBOOT-FILE WIRD ANGELEGT
     ."                                                 <92H>
2410 SLOW : PRINT : PRINT CHR$(159)"WOLLEN SIE AB
     SPEICHERN ?";: GOSUB 2550                          <R24>
2420 IF A$="N" THEN CLOSE 2: CLOSE 1: GOTO 2150         <OAT>
2430 PRINT#1,"B-F 0 1 0": PRINT#1,"B-F 0 1 1": PR
     INT : N$=""                                        <PE0>
2440 PRINT "{DOWN}EINSTELLUNGEN WERDEN GESPEICHER
     T : PRINT : FAST                                   <RK5>
2450 PRINT#1,"B-P 13 0": FOR I=0 TO 255: PRINT#2,
     CHR$(B1(I));: NEXT                                 <OII>
2460 PRINT#1,"U2 13 0 1 0": PRINT "BLOCK 1: "DS$        <O0L>
2470 IF FT THEN 2500                                    <3D3>
2480 PRINT#1,"B-P 13 0": FOR I=0 TO 255: PRINT#2,
     CHR$(B2(I));: NEXT                                 <QAI>
2490 PRINT#1,"U2 13 0 1 1": PRINT "BLOCK 2: "DS$        <3J5>
2500 PRINT#1,"B-A 0 1 0": PRINT#1,"B-A 0 1 1": PR
     INT "BELEGEN: "DS$                                 <625>
2510 CLOSE 3: CLOSE 2: CLOSE 1: SLOW : END              <0C3>
2520 IF BR THEN IF ER=30 THEN RESUME NEXT               <00G>
2530 PRINT : PRINT "{DOWN}?" ERR$(ER)" ERROR IN"E
     L CHR$(7): CLOSE 2: CLOSE 1: CLOSE 3               <J19>
2540 SLOW : HELP : END                                  <N3S>
2550 GET KEY A$: IF A$<>"J" AND A$<>"N" THEN 2550       <TM7>
2560 L=PEEK(241): IF A$="J" THEN PRINT "{CTRL+G,W
     HITE} JA": ELSE PRINT "{CTRL+G,WHITE} NEIN"        <QLO>
2570 POKE 241,L: RETURN                                 <F6H>
2580 Z=PEEK(235): S=PEEK(236)                           <C25>
2590 WINDOW S,Z,S+L-1,Z,1: PRINT CHR$(27)"M"            <V7V>
2600 INPUT#3,A$: IF A$="" THEN 2600                     <NSG>
2610 WINDOW 0,0,39,24: POKE 235,Z: PRINT CHR$(27)
     "L": RETURN                                        <CQR>
2620 :                                                  <001>
2630 REM DATEN FUER BLOCK 1                             <3SF>
2640 :                                                  <30D>
2650 DATA 43,42,4D,00,10,00,01,EA,EA,EA,EA,EA,EA,
     EA,EA,EA                                           <UP8>
2660 DATA EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,
     EA,EA,EA                                           <R58>
2670 DATA EA,EA,EA,EA,EA,EA,EA,EA,EA,A9,49,8D,00,
     0A,A9,0B                                           <BKL>
2680 DATA 8D,01,0A,EA,EA,EA,EA,EA,A2,04,86,D0,CA,
     BD,AF,0B                                           <EAG>
2690 DATA 9D,4A,03,CA,10,F7,4C,4F,0B,20,4F,0B,4C,
     03,40,A9                                           <F2E>
2700 DATA 00,8D,20,D0,A9,00,8D,21,D0,A9,01,85,F1,
     24,D7,10                                           <M1M>
2710 DATA 09,A9,07,85,F1,A9,01,8D,30,D0,60,A2,08,
     BD,A6,0B                                           <IBR>
2720 DATA 9D,00,80,CA,10,F7,A9,F8,85,FC,A9,FF,85,
     FD,A9,FC                                           <1NM>
2730 DATA 8D,B9,02,A9,98,A2,01,A0,00,20,77,FF,A9,
     0B,A2,01                                           <NFO>
2740 DATA A0,01,20,77,FF,4C,4F,0B,EE,20,D0,EE,21,
     D0,A2,E8                                           <D9I>
2750 DATA CA,D0,FD,4C,98,0B,98,0B,00,00,C3,C2,CD,
     38,30,52                                           <9FP>
2760 DATA 55,4E,0D                                      <E2N>
```

Listing 1. »Uniboot« bitte mit dem Checksummer 128 (Seite 158) eingeben (Schluß)

Schnelles Sortieren mit dem C 128

3000 Strings in 2 Sekunden sortiert – eine superschnelle Sortierroutine für den C 128, die in Maschinensprache geschrieben ist.

Wer die Literatur zum C128 aufmerksam verfolgt, wird feststellen, daß noch nirgendwo eine schnelle Sortierroutine in Assembler für diesen Computer veröffentlicht wurde. Der Grund, warum noch keine derartige Routine vom C64 auf den C128 umgesetzt wurde (für den C64 gibt es bereits eine Unzahl eleganter Programme zum schnellen Sortieren), ist in der Schwierigkeit zu suchen, die der C128 dem Programmierer durch seine Bankverwaltung stellt.

3000 Strings in 2 Sekunden sortiert

Wir wollen dieses Manko beseitigen und liefern Ihnen mit »Quicksort C128« eine Sortierroutine, die nur zirka zwei Sekunden zum Sortieren von 3000 Strings benötigt.

Im Rahmen des Kurses »Effektives Programmieren« veröffentlichten wir in der Ausgabe 7/86 des 64'er-Magazins ein rekursiv arbeitendes Quicksort-Maschinenprogramm, das den Prozessorstapel verwendet und sehr schnell arbeitet. Dieses Programm war die Basis für Quicksort 128. Wichtig bei der Umsetzung war auch, daß man die Sortierroutine auch noch nachträglich – beispielsweise aus einem bereits laufenden Basic-Programm – laden und starten kann.

Zwei Probleme sind es vor allem, die bei der Umsetzung der C64-Routine einem schnellen Erfolg im Weg liegen: eine andere Art der Speicherung von String-Array-Elementen beim C128 und die Bankstruktur dieses Computers. Sehen wir uns die Lösungen der Reihe nach an.

Die String-Arrays im C 128

Ein beispielsweise durch DIM A$(1000) definiertes String-Array erzeugt in der Bank 1 des C128 drei Arten von Einträgen. Am Anfang eines String-Arrays steht der Array-Kopf (oder auch Header genannt), der dem im C64 erzeugten gleich ist (siehe Tabelle 1).

An diesen Kopf schließen sich die einzelnen Array-Elemente an. Beim String-Array sind das jeweils 3 Byte, die man String-Descriptoren nennt (was etwa String-»Beschreibungen« bedeutet). Tabelle 2 zeigt Ihnen den Aufbau eines solchen String-Deskriptors. Wer den C64 genau kennt, stellt den identischen Aufbau fest. Der dritte Eintrag ist nun der jeweilige Text, dessen Länge und Adresse im Descriptor angegeben ist. Tabelle 3 erläutert diesen Texteintrag.

Die in der Tabelle 3 zuletzt genannten zwei Byte gibt es beim C64 nicht. Durch sie wird eine unangenehme Verzögerung bei der Stringverarbeitung, die sogenannte »Garbage Collection«, erheblich beschleunigt. Einer der Tricks bei Sortierprogrammen ist es, zum Ordnen der Texte nicht die Texte selbst, sondern die dazugehörigen Descriptoren zu vertauschen. Beim C128 genügt das aber – wegen der beiden zusätzlichen Bytes – nicht. Man muß auch diese Code-Descriptoren auswechseln. Die im Descriptor angegebene String-Länge umfaßt nur die ASCII-Codes des Textes. Addiert man diese String-Länge zur String-Adresse (Byte 2 und 3 im Descriptor), dann erhält man einen Zeiger auf diese Code-Descriptoren. In den Zeilen 3180 bis 3450 des beigefügten Quelltextes (Listing 1) finden Sie diese Berechnung und die Vertauschung der Code-Descriptoren.

Bankprobleme

Von den Problemen mit der Bankverwaltung weiß ein jeder C128-Assembler-Programmierer Klagelieder zu singen: Wo legt man am besten ein Maschinenprogramm ab, damit es fehlerfrei arbeitet? Hier ein kurzer Überblick:

In der Bank 0 liegt der Basic-Text im Bereich von $1C00 bis $FF00. In Bank 1 finden sich alle Variablen; Arrays und Strings liegen im Bereich zwischen $0400 und $FF00. Der Basic-Interpreter, das Betriebssystem und die Ein- und Ausgabeeinheiten finden sich in Bank 15 von Adresse $4000 an aufwärts. Im Einschaltzustand liegt folgende Konfiguration vor:

Von $0000 bis $3FFF ist Bank 0 und von $4000 bis $FFFF ist Bank 15 aktiviert. Viel freier Platz für Maschinenprogramme (zirka 2,3 KByte) findet sich unterhalb des Basic-Bereichs in Bank 0 von $1300 bis $1BFF.

Keine Probleme entstehen für kurze Assembler-Programme, die sich mit diesem Speicherbereich in der Bank 0 zufriedengeben. Ebenfalls möglich ist der Zugriff aus einem Maschinenprogramm auf Routinen der Bank 15 (weil diese ja ab $4000 eingeblendet ist). Aber bei einem Zugriff auf Bank 1 – also die Variablen – wachsen plötzlich ungeahnte Schwierigkeiten hervor: Die Bank 1 müßte eingeschaltet werden. Dann vergißt der Computer aber das Programm in der Bank 0. Man muß also in die Bank 1 irgendwie eine anzuspringende Routine packen, die dann wieder die Bank 0 einschaltet. Wohin aber mit dieser Routine, wenn man nicht massiv in den Speicher eingreifen will (von $0400 an bis $FF00 liegen ja die Variablen)? Nun gibt es die sogenannte Common-Area zwischen $0000 und

Byte	Inhalt		
Byte 1	Array-Name	1. Buchstabe (ASCII-Code)	
Byte 2	2. Buchstabe + Kennung	(ASCII-Code+128)	
Byte 3	Array-Länge	als 16-Bit-Wert,	LSB
Byte 4			MSB
Byte 5	Anzahl Dimensionen	8-Bit-Wert	
Byte 6	Anzahl Elemente in letzter Dimension	16-Bit-Wert,	LSB
Byte 7			MSB
Byte 8 ff.	Anzahl Elemente in vorletzter Dimension und so weiter		LSB

Tabelle 1. Aufbau des Array-Kopfes (String-Array)

Byte	Inhalt	
Byte 1	String-Länge	
Byte 2	Adresse des Textes	LSB
Byte 3	Adresse des Textes	MSB

Tabelle 2. Der Aufbau eines String-Descriptors

Byte	Inhalt	
Byte 1 bis n	ASCII-Codes der Zeichen des Textes	
Byte n+1	Adresse des zum Texteintrag gehörenden Descriptors,	LSB
Byte n+2		MSB

Tabelle 3. Der Aufbau eines Textfeldes

$0400. Das ist ein Speicherbereich in der Bank 0, der von allen anderen Bänken aus zugänglich ist und von dem aus man auf alle anderen Bänke zugreifen kann. Leider ist diese Common-Area (»gemeinsamer Bereich«) aber randvoll mit der erweiterten Zeropage, ein Programm hat hier keinen Platz mehr. Was tun?

Zunächst gibt es die Möglichkeit, die Common-Area so zu vergrößern, daß das von uns benötigte Programm darin Platz hat. Ein MMU-Register erledigt das, und einige weitere Zeiger müssen verändert werden (Basic-Start, Variablenzeiger, Bildschirm-Adresse, etc.).

Dieses Vorgehen scheitert aber dann, wenn man – wie wir es hier benötigen – das Maschinenprogramm aus einem laufenden Basic-Programm heraus laden und starten will. Es sei denn, man gewöhnt sich an, an den Anfang jedes Programms grundsätzlich diese Speicher-Umkonfigurationen zu legen. Das schafft aber einen unnötigen Programmaufwand, der vermieden werden sollte.

Nützliche Betriebssystem-Routinen

Eine andere Möglichkeit ist die Verwendung bestimmter Routinen des Betriebssystems (FETCH, STASH, etc.) für die Kommunikation zwischen den Bänken. Im Prinzip verlangsamen diese Routinen zwar den Ablauf der Programme etwas, ihre Verwendung ist aber häufig unvermeidbar. Im vorliegenden Fall aber, bei dem der Prozessorstapel als Zähler für die Rekursionstiefe dient, wird der Gebrauch dieser verschiedenen – mittels JSR aufzurufenden – Routinen zum gefährlichen Spiel. Jeder JSR-Befehl wirkt sich auf den Stapel aus, was zu Abstürzen der Sortierroutine führen kann.

Man hätte natürlich das ganze Quicksort-Programm noch in die Bank 1 legen können. Auch das entfällt wegen der damit nötigen Veränderungen der Speicherorganisation (wie gesagt: Das Programm sollte aus einem laufenden Basic-Programm heraus...).

Die Lösung

Die Lösung erscheint auf den ersten Blick etwas ungewöhnlich: Das Programm wird in den Speicherbereich ab $1300 in Bank 0 geladen. Nach den Start beginnt eine Verschieberoutine, den Qicksort-Teil Byte für Byte in die Common-Area (ab $0200) zu verschieben. Gleichzeitig wandert der Inhalt der Common-Area in den zuvor vom Quicksort-Programm belegten Speicher: Es findet also ein Austausch statt. Nach erfolgter Verlagerung startet das nun in der Common-Area liegende Programm, schaltet die Bank 1 ein, sortiert das erste definierte Array, schaltet wieder die normale Speicherkonfiguration ein und sorgt für einen erneuten Austausch.

Weil die per Betriebssystem festgelegten Interrupt-Bearbeitungen Speicherstellen der Common-Area verändern, bleiben die Unterbrechungen vom Programmstart bis zu seinem Ende unterbunden.

Listing 2 zeigt »Quicksort 128«, das Sie bitte mit dem MSE eingeben (beachten Sie dazu Seite 158). In Listing 1 finden Sie den mit dem Top-Ass erstellten Quelltext für Quicksort 128. Listing 3 zeigt ein kleines Basic-Testprogramm, das den Gebrauch der Sortierroutine zeigen soll.

Noch einige Hinweise: Das zu sortierende String-Array muß als erstes Array definiert sein. Quicksort ordnet es dann in aufsteigender Reihenfolge. Gestartet wird die Sortierroutine mit

```
SYS DEC("1300")
```

Von einem Programm aus kann Quicksort 128 einfach mit dem Befehl

```
BLOAD"QUICKSORT 128"
```

nachgeladen werden. 3000 Strings alphabetisch zu sortieren dauert etwa 2 Sekunden, was Sie mit dem Basic-Testprogramm (Listing 3) ausprobieren können. Dieses Testprogramm erzeugt per Zufall die gewünschte Anzahl Strings (zu je drei Buchstaben) in einem Array, zeigt sie auf dem Bildschirm, sortiert sie dann und zeigt das sortierte Feld erneut an. Eine Zeitangabe für die Sortierdauer findet statt, ist aber wegen des abgeschalteten Interrupts während des Sortierens mit Vorsicht zu genießen.

(Heimo Ponnath/Martin Jobst)

```
ready.
10   -;******************************************************
20   -;*                                                    *
30   -;*           c 128  quicksort - routine               *
40   -;*                                                    *
50   -;*          heimo ponnath  hamburg  1986              *
60   -;* nach einer c64-routine von k.soerensen  64er 7(86) *
70   -;*                                                    *
80   -;******************************************************
90   -;
100  -                    .base $1300
110  -                    .code 0,$9300
120  -;
130  -;progr. startet bei $1300, wird aber bei assemblierung ab $9300 abgelegt
140  -; um nicht den quelltext oder den topass zu zerstoeren.
150  -; der code muss per monitor nach $1300 geschoben werden.
160  -;
170  -;----- hilfszellen ------------------------------
180  -;
190  -                    .define xadr= $0a;adresse von a$(x) in $0a/0b
200  -                    .define yadr= $0c;adresse von a$(y) in $0c/0d
210  -                    .define vadr= $0e;adresse von a$(v) in $0e/0f
220  -                    .define xdes= $10;descriptor a$(x) in $10,11,12
230  -                    .define ydes= $13;descriptor a$(y) in $13,14,15
240  -                    .define vdes= $16;descriptor a$(v) in $16,17,18
250  -                    .define lg= $19 ;linke grenze in $19,1a
260  -                    .define rg= $1b ;rechte grenze in $1b,1c
270  -                    .define verl= $1d;laengenvergleich
280  -                    .define zwsp= $1e;zwischenspeicher (1 byte)
290  -                    .define zwsp1= $1f;zwischenspeicher (2 byte)
300  -                    .define zwsp2= $21;zwischenspeicher (2 byte)
310  -;
320  -                    .define arytab= $31;feldanfang in bank 1
330  -;
335  -                    .define st1= $fb;umladevektor
340  -                    .define stin= $fc;stapel eingangswert
350  -                    .define stmi= $fd;stapelminimum,umladevektor
360  -                    .define stug= $fe;stapeluntergrenze
370  -;
380  -                    .define zero= $200;hier faengt das verschobene progra
mm an
385  -                    .define retour= $03db;endadresse im verschobenen prog
ramm
390  -;
400  -                    .define cr= $ff00;konfigurationsregister
410  -                    .define lcrb= $ff02;ladekontrollregister b (enthaelt
$7f)
420  -;
430  -;
440  -;
450  -;
460  -;****** steuerprogramm quicksort ******************
470  -;
475  -steuer0            sei             ;unterbrechungen verhindern
480  -                  jsr umlad       ;programm umladen in common area
490  -                  lda #<(zero)    ;indirekten sprungvektor
500  -                  sta st1         ;auf neuen programmanfang richten
510  -                  lda #>(zero)
520  -                  sta st1+1
530  -                  jmp (st1)       ;sprung ins umgeladene quicksortprogra
mm
540  -leave              jsr umlad       ;common area restaurieren, programm zu
rueckholen
545  -                  cli             ;unterbrechungen freigeben
550  -                  rts             ;zurueck ins basicprogramm
560  -;
570  -;--- unterprogramm zum umladen von commonbereich und programm ---
580  -;
590  -umlad              lda #<(zero)    ;vektor quellbereich belegen
600  -                  sta st1         ;lsb
610  -                  lda #>(zero)
620  -                  sta st1+1       ;msb
630  -                  lda #<(anfang)  ;vektor zielbereich belegen
640  -                  sta stmi        ;lsb
650  -                  lda #>(anfang)
660  -                  sta stmi+1      ;msb
670  -;
680  -                  ldx #$01        ;anzahl pages -1 (512 bytes umladen)
690  -umlad1             ldy #$00        ;offset initialisieren
700  -umlad2             lda (st1),y     ;byte aus quellbereich lesen
710  -                  pha             ;beiseite legen
720  -                  lda (stmi),y    ;byte aus zielbereich lesen
730  -                  sta (st1),y     ;und in quellbereich schreiben
740  -                  pla             ;byte wieder hervorholen
750  -                  sta (stmi),y    ;und in zielbereich schreiben
760  -                  iny             ;naechstes byte
770  -                  bne umlad2      ;bis 256 bytes vertauscht sind
780  -                  inc st1+1       ;msb quellbereich erhoehen
790  -                  inc stmi+1      ;und msb zielbereich ebenfalls
800  -                  dex             ;pagezaehler herunterzaehlen
810  -                  beq umlad2      ;weiter bis unterlauf eintritt
820  -                  rts             ;zurueck ins steuerprogramm
830  -;
840  -;
```

```
850  -;------- zu verschiebendes programm ----------------
860  -;
870  -;--- speicherkonfiguration umschalten auf bank 1 (alles ram) ---
880  -;
890  -anfang                 sta lcrb       ;dadurch wird der inhalt des pcrb ins
cr gesetzt
900  -;
910  -;
920  -;----- programm quicksort ---------------------
930  -;
940  -;--- teil 1: retten der basic-daten -----------
950  -;
960  -;- stapelzeiger -
970  -start                  tsx            ;stapelzeiger ins x-register
980  -                       stx stin       ;und abspeichern
990  -                       stx stmi
1000 -                       txa            ;freien platz im stack sichern
1010 -                       sec
1020 -                       sbc #$52
1030 -                       bcs spr        ;genug platz:weiter, sonst
1040 -                       jmp retour     ;abbruch des programmes
1050 -spr                    adc #$02       ;stapeluntergrenze
1060 -                       sta stug       ;festlegen
1070 -;- zeropageinhalte sichern -
1080 -                       ldx #$19       ;zaehler initialisieren
1090 -zpweg                  lda $0a,x      ;umladeschleife
1100 -                       pha            ;werte auf stapel legen
1110 -                       dex
1120 -                       bpl zpweg      ;bis alle inhalte gesichert
1130 -                       tsx            ;stapelzeiger als eingangs-
1140 -                       stx stin       ;wert festhalten
1150 -;
1160 -;--- teil 2: festlegen anfangsbedingungen -----
1170 -;
1180 -;- schachtelungstiefe z=0 -
1190 -sort                   lda #$00
1200 -                       pha
1210 -                       pha
1220 -;- linke grenze lg=1 -
1230 -                       clc            ;element a$(1) im
1240 -                       lda arytab     ;ersten definierten
1250 -                       adc #$0a       ;array feststellen
1260 -                       sta lg         ;und adresse in
1270 -                       lda arytab+1   ;lg und lg+1
1280 -                       adc #$00       ;eintragen
1290 -                       sta lg+1       ;a$(0) wird uebersprungen
1300 -;- rechte grenze rg=a -
1310 -                       clc            ;letztes element a$(a)
1320 -                       ldy #$02       ;durch addieren der feld-
1330 -                       lda (arytab),y ;laenge zum arraystart
1340 -                       adc arytab     ;suchen
1350 -                       tax
1360 -                       iny
1370 -                       lda (arytab),y
1380 -                       adc arytab+1
1390 -                       tay
1400 -                       sec            ;laenge des letzten descriptors
1410 -                       txa            ;abziehen um die start-
1420 -                       sbc #$03       ;adresse von a$(a) zu
1430 -                       sta rg         ;ermitteln und nach
1440 -                       pha            ;rg/rg+1 und auf den
1450 -                       tya            ;stapel zu schieben.
1460 -                       sbc #$00
1470 -                       sta rg+1
1480 -                       pha
1490 -                       bne vstring    ;hier immer verzweigung
1500 -;- verschachtelungstiefe z=1 -
1510 -;
1520 -bruecke4               bne sort
1530 -;
1540 -;--- teil 3: vergleichsstring berechnen -------
1550 -;
1560 -;- x=lg, y=rg -
1570 -;
1580 -vstring                ldx #$03       ;inhalte von lg und rg
1590 -ladxy                  lda lg,x       ;in xadr und yadr
1600 -                       sta xadr,x     ;umladen
1610 -                       dex
1620 -                       bpl ladxy
1630 -;
1640 -;- v=x+y -
1650 -;
1660 -                       clc            ;addieren startadressen des
1670 -                       lda xadr       ;ersten und des
1680 -                       adc yadr       ;letzten descriptors
1690 -                       tax
1700 -                       lda xadr+1
1710 -                       adc yadr+1
1720 -;
1730 -;- v=int(v/2) -
1740 -;
1750 -                       lsr            ;msb der summe rechts shiften =
1760 -                       sta vadr+1     ;division durch 2
1770 -                       txa            ;lsb der summe rechts rotieren
1780 -                       ror
1790 -                       bcc spr1       ;summe gerade,dann sprung
1800 -                       sbc #$01       ;sonst begradigen
1810 -                       bcs spr1
1820 -                       dec vadr+1     ;wenn unterlauf msb korrigieren
1830 -spr1                   sta vadr
1840 -;
1850 -;- vergleichsstring v$=a$(v) -
1860 -;
1870 -                       ldy #$00       ;descriptor von v$ uebertragen
1880 -                       lda (vadr),y   ;in vdes
1890 -                       sta vdes
1900 -                       iny
1910 -                       lda (vadr),y
1920 -                       sta vdes+1
1930 -                       iny
1940 -                       lda (vadr),y
1950 -                       sta vdes+2
1960 -;
1970 -;--- teil 4: stringvergleich linker teil --------
1980 -;
1990 -;- string links von v$: x$=a$(x)
2000 -;
2010 -verglx                 ldy #$00       ;descriptor von x$ uebertragen
2020 -                       lda (xadr),y   ;in xdes
2030 -                       sta xdes
2040 -                       iny
2050 -                       lda (xadr),y
2060 -                       sta xdes+1
2070 -                       iny
2080 -                       lda (xadr),y
2090 -                       sta xdes+2
2100 -;- stringlaengenvergleich -
2110 -                       ldx #$00       ;wenn x$ kuerzer als v$:
2120 -                       lda xdes       ;     x = 0 ,  verl = len(x$)
2130 -                       cmp vdes       ;wenn v$ kuerzer x$:
2140 -                       bcc spr2       ;     x    = 1
2150 -                       inx            ;     verl = len(v$)
2160 -                       lda vdes
2170 -spr2                   sta verl
2180 -;
2190 -;- stringvergleich : x$ < oder > v$
2200 -;
2210 -                       ldy #$00       ;offset initialisieren
2220 -verglx1                lda (xdes+1),y ;zeichen holen
2230 -                       cmp (vdes+1),y ;und vergleichen
2240 -                       bne verglx2    ;zeichen ungleich: sprung
2250 -                       iny
2260 -                       cpy verl       ;schon stringlaenge erreicht?
2270 -                       bcc verglx1    ;nein,naechstes zeichen
2280 -                       cpx #$01
2290 -verglx2                bcs vergly
2300 -;
2310 -;- naechstes element vergleichen: x=x+1
2320 -;
2330 -                       clc            ;erhoehen von xadr um 3
2340 -                       lda xadr       ;fuer den naechsten
2350 -                       adc #$03       ;stringdescriptor
2360 -                       sta xadr
2370 -                       bcc verglx     ;weiter zum naechsten vergleich
2380 -                       inc xadr+1
2390 -                       bcs verglx
2400 -;
2410 -;- sprung ------------
2420 -bruecke5               bne bruecke4
2430 -bruecke1               bcc vstring
2440 -;
2450 -;--- teil 5: stringvergleich rechter teil ----------
2460 -;
2470 -;- string rechts von v$ : y$=a$(y)
2480 -;
2490 -vergly                 ldy #$00       ;descriptor von y$ uebertragen
2500 -                       lda (yadr),y   ;in ydes
2510 -                       sta ydes
2520 -                       iny
2530 -                       lda (yadr),y
2540 -                       sta ydes+1
2550 -                       iny
2560 -                       lda (yadr),y
2570 -                       sta ydes+2
2580 -;- stringlaengenvergleich -
2590 -                       ldx #$00       ;wenn v$ kuerzer als y$:
2600 -                       lda vdes       ;     x = 0 , verl = len(v$)
2610 -                       cmp ydes       ;wenn y$ kuerzer als v$:
2620 -                       bcc spr3       ;     x    = 1
2630 -                       inx            ;     verl = len(y$)
2640 -                       lda ydes
2650 -spr3                   sta verl
2660 -;
2670 -;- stringvergleich : y$ < oder > v$
2680 -;
2690 -                       ldy #$00       ;offset initialisieren
2700 -vergly1                lda (vdes+1),y ;zeichen holen
2710 -                       cmp (ydes+1),y ;und vergleichen
2720 -                       bne vergly2    ;zeichen ungleich:sprung
2730 -                       iny
2740 -                       cpy verl       ;schon stringlaenge erreicht?
2750 -                       bcc vergly1    ;nein,naechstes zeichen
2760 -                       cpx #$01
2770 -vergly2                bcs tausch
2780 -;
2790 -;- naechstes element vergleichen: y=y-1
2800 -;
2810 -                       sec            ;yadr um 3 bytes
2820 -                       lda yadr       ;erniedrigen fuer den
2830 -                       sbc #$03       ;naechsten
2840 -                       sta yadr       ;stringdescriptor
2850 -                       bcs vergly     ;weiter zum naechsten vergleich
2860 -                       dec yadr+1
2870 -                       bcc vergly
2880 -;
2890 -;- sprung -----------
2900 -bruecke2               bcs verglx
2910 -bruecke3               bcc bruecke1
2920 -bruecke6               bne bruecke5
2930 -;
2940 -;--- teil 6: strings vertauschen -----------
2950 -;
2960 -;-pruefen ob x >= y -
2970 -;
2980 -tausch                 lda yadr+1 .   ;msb vergleichen
2990 -                       cmp xadr+1
3000 -                       bcc bruecke8   ;wenn x>y dann sprung
3010 -                       bne swap       ;wenn x<y dann vertauschen
3020 -                       lda xadr       ;lsb vergleichen
3030 -                       cmp yadr
3040 -                       bcs bruecke9   ;wenn x >= y dann sprung
3050 -;
3060 -;- vertauschen der string- und codedescriptoren -
3070 -;- stringdescriptoren: -
3080 -swap                   ldx #$02
3090 -                       ldy #$02
3100 -swap1                  lda xdes,x     ;x-descriptor
3110 -                       sta (yadr),y   ;nach A$(Y)
3120 -                       lda ydes,x     ;y-descriptor
3130 -                       sta (xadr),y   ;nach A$(X)
3140 -                       dex
3150 -                       dey
3160 -                       bpl swap1
3170 -;
3180 -;- codedescriptoren: -
3190 -;
3200 -                       clc            ;berechnen der orte der codedescriptor
en
3210 -                       lda xdes+1     ;lsb x-string
3220 -                       adc xdes       ;plus stringlaenge
3230 -                       sta zwsp1      ;ergibt lsb des ortes codedescriptor x
3240 -                       lda xdes+2     ;msb x-string
3250 -                       adc #$00       ;eventuell carry addieren
3260 -                       sta zwsp1+1    ;msb ort codedescriptor x
3270 -;
3280 -                       clc            ;dasselbe fuer den y-string
3290 -                       lda ydes+1     ;lsb
3300 -                       adc ydes       ;+ stringlaenge
3310 -                       sta zwsp2      ;lsb zeiger auf codedescriptor y
3320 -                       lda ydes+2     ;msb
3330 -                       adc #$00
3340 -                       sta zwsp2+1    ;msb des zeigers
3350 -;
3360 -                       ldy #$01       ;zaehler initialisieren
3370 -swap2                  lda (zwsp1),y  ;codedescriptor x laden
3380 -                       sta zwsp       ;und beiseite legen
3390 -                       lda (zwsp2),y  ;codedescriptor y laden
3400 -                       sta (zwsp1),y  ;und nach x schieben
3410 -                       lda zwsp       ;x wieder hervorholen
3420 -                       sta (zwsp2),y  ;und nach y schieben
3430 -                       dey            ;herunterzaehlen
3440 -                       beq swap2      ;bis unterlauf des zaehlers
3450 -;
3460 -;- naechste descriptoradresse x=x+1
3470 -;
3480 -                       clc            ;zur aktuellen adresse
3490 -                       lda xadr       ;des x-string
3500 -                       adc #$03       ;3 byte addieren fuer
3510 -                       sta xadr       ;neue descriptoradresse x
3520 -                       bcc spr4
3530 -                       inc xadr+1
3540 -;
3550 -;-naechste descriptoradresse y=y-1
3560 -;
3570 -spr4                   sec            ;von der aktuellen adresse
3580 -                       lda yadr       ;des y-string
3590 -                       sbc #$03       ;3 byte subtrahieren fuer
```

Listing 1. Die schnelle Sortierroutine Quicksort 128, hier als Top-Ass-Quelltext.

```
3600 -                      sta yadr        ;neue descriptoradresse y
3610 -                      bcs spr5
3620 -                      dec yadr+1
3630 -;
3640 -;-falls x <= y -
3650 -;
3660 -spr5                  lda yadr+1      ;msb y-descriptoradresse vergleichen
3670 -                      cmp xadr+1      ;mit msb x-descriptoradresse
3680 -                      bcc zstufen     ;wenn x > y dann sprung
3690 -                      bne bruecke2    ;wenn x < y dann sprung
3700 -                      lda yadr        ;lsb y-descriptoradresse vergleichen
3710 -                      cmp xadr        ;mit lsb x-descriptoradresse
3720 -                      bcs bruecke2    ;wenn x < y dann sprung
3730 -                      bcc zstufen     ;hier immer sprung
3740 -;
3750 -;- sprung ------------
3760 -bruecke10             bcc bruecke3
3770 -bruecke7              bne bruecke6
3780 -bruecke8              bcc zstufen
3790 -bruecke9              bcs zstufen
3800 -;
3810 -;--- teil 7: naechste schachtelungstiefe ermitteln ------
3820 -;
3830 -;- z=z+1,rg=y -
3840 -zplus                 lda rg          ;rechte grenze auf den
3850 -                      pha             ;stapel schieben:
3860 -                      lda rg+1        ;bedeutet z=z+1
3870 -                      pha
3880 -                      lda yadr        ;neue rechte grenze
3890 -                      sta rg          ;ist descriptoradresse von y
3900 -                      lda yadr+1
3910 -                      sta rg+1
3920 -                      clc             ;hier erfolgt immer sprung
3930 -                      bcc bruecke10   ;nach vstring (neuer vergleichsstring)
3940 -;
3950 -;- lg = lg+1 -
3960 -zgleich               clc             ;zur linken grenze 3 bytes
3970 -                      lda lg          ;addieren fuer den
3980 -                      adc #$03        ;naechsten descriptor
3990 -                      sta lg
4000 -                      bcc spr6
4010 -                      iny
4020 -spr6                  sty lg+1
4030 -                      cpy rg+1        ;msb der grenzadressen vergleichen
4040 -                      bcc bruecke10   ;wenn rg > lg sprung nach vstring
4050 -                      bne zminus      ;wenn rg < lg sprung nach zminus
4060 -                      cmp rg          ;lsb grenzadressen vergleichen
4070 -                      bcc bruecke10   ;wenn rg > lg sprung nach vstring
4080 -                      bcs zminus      ;sonst sprung nach zminus
4090 -;
4100 -zstufen               lda lg          ;vergleich linke grenze mit
4110 -                      ldy lg+1        ;y-descriptoradresse
4120 -                      cpy yadr+1      ;msb-vergleich
4130 -                      bcc stack       ;wenn yadr > lg dann sprung
4140 -                      bne zgleich     ;wenn yadr < lg dann sprung
4150 -                      cmp yadr        ;sonst lsb-vergleich
4160 -                      bcc stack       ;wenn yadr > lg dann sprung
4170 -                      bcs zgleich     ;sonst sprung nach zgleich
4180 -;
4190 -;- stapel pruefen -
4200 -stack                 tsx             ;aktuellen stapelzeiger vergleichen
4210 -                      cpx stmi        ;mit stapelminimum
4220 -                      bcs zplus       ;wenn stmi <= stapelzeiger sprung
4230 -                      stx stmi        ;neues stapelminimum
4240 -                      cpx stug        ;vergleich mit stapeluntergrenze
4250 -                      bcs zplus       ;wenn stug <= stapelzeiger dann sprung
4260 -                      ldx stin        ;sonst ruecksprung
4270 -                      txs
4280 -                      bne bruecke7    ;zum anfang (sort)
4290 -;
4300 -;- z = z-1 -
4310 -zminus                pla             ;letzte rechte grenze
4320 -                      sta rg+1        ;vom stapel holen:
4330 -                      pla             ;bedeutet
4340 -                      sta rg          ;z=z-1
4350 -                      ldx rg+1        ;pruefen, ob
4360 -                      cpx #$00        ;msb rechte grenze
4370 -                      bne zgleich     ;ungleich null,dann sprung
4380 -;
4390 -;--- teil 8: rueckkehr ins basic ---------------
4400 -;
4410 -;- stapelzeiger -
4420 -aus                   ldx stin
4430 -                      txs
4440 -;
4450 -;- zeropage zurueckholen -
4460 -                      ldx #$00        ;zaehler initialisieren
4470 -zprueck               pla             ;wert von stapel holen
4480 -                      sta $0a,x       ;und in die zeropagestelle schreiben
4490 -                      inx             ;naechster wert
4500 -                      cpx #$1a
4510 -                      bcc zprueck
4520 -;
4530 -;--- speicherkonfiguration umschalten auf normalkonfiguration ---
4540 -;
4550 -ende                  lda #$00        ;konfigurationsbyte fuer normalzustand
4560 -                      sta cr          ;ins cr schreiben
4570 -;
4580 -;--- verlassen und zurueckladen des programmes ---------------
4590 -;
4600 -                      lda #<(leave)   ;indirekten sprungvektor laden (lsb)
4610 -                      sta zwsp1
4620 -                      lda #>(leave)   ;msb
4630 -                      sta zwsp1+1
4640 -                      jmp (zwsp1)     ;indirekter sprung ins steuerprogramm
4650 -;
4660 -;--- ende des zu verschiebenden programmteiles ---------------
4670 -;
```

Listing 1. »Quicksort 128«-Quelltext (Schluß)


```
Name : prog 1 obj 1300    1300 1529
-------------------------------------
1300 : 78 20 14 13 a9 00 85 fb  98
1308 : a9 02 85 fc 6c fb 00 20  9a
1310 : 14 13 58 60 a9 00 85 fb  79
1318 : a9 02 85 fc a9 3d 85 fd  5a
1320 : a9 13 85 fe a2 01 a0 00  49
1328 : b1 fb 48 b1 fd 91 fb 68  4c
1330 : 91 fd c8 d0 f3 e6 fc e6  44
1338 : fe ca f0 ec 60 8d 02 ff  ef
1340 : ba 86 fc 86 fd 8a 38 e9  36
1348 : 52 b0 03 4c db 03 69 02  bc
1350 : 85 fe a2 19 b5 0a 48 ca  82
1358 : 10 fa ba 86 fc a9 00 48  12
1360 : 48 18 a5 31 69 0a 85 19  73
1368 : a5 32 69 00 85 1a 18 a0  4c
1370 : 02 b1 31 65 31 aa c8 b1  33
1378 : 31 65 32 a8 38 8a e9 03  83
1380 : 85 1b 48 98 e9 00 85 1c  a5
1388 : 48 d0 02 d0 d0 a2 03 b5  6d
1390 : 19 95 0a ca 10 f9 18 a5  cc
1398 : 0a 65 0c aa a5 0b 65 0d  10
13a0 : 4a 85 0f 8a 6a 90 06 e9  d9
13a8 : 01 b0 02 c6 0f 85 0e a0  f1
13b0 : 00 b1 0e 85 16 c8 b1 0e  48
13b8 : 85 17 c8 b1 0e 85 18 a0  e0
13c0 : 00 b1 0a 85 10 c8 b1 0a  ee
13c8 : 85 11 c8 b1 0a 85 12 a2  98
13d0 : 00 a5 10 c5 16 90 03 e8  23
13d8 : a5 16 85 1d a0 00 b1 11  80
13e0 : d1 17 d0 07 c8 c4 1d 90  9a
13e8 : f5 e0 01 b0 11 18 a5 0a  20
13f0 : 69 03 85 0a 90 c9 e6 0b  86
13f8 : b0 c5 d0 8f 90 8f a0 00  b9
1400 : b1 0c 85 13 c8 b1 0c 85  d0
1408 : 14 c8 b1 0c 85 15 a2 00  fa
1410 : a5 16 c5 13 90 03 e8 a5  a4
1418 : 13 85 1d a0 00 b1 17 d1  d7
1420 : 14 d0 07 c8 c4 1d 90 f5  db
1428 : e0 01 b0 13 38 a5 0c e9  cc
1430 : 03 85 0c b0 c9 c6 0d 90  37
1438 : c5 b0 84 90 bf d0 bb a5  45
1440 : 0d c5 0b 90 6d d0 06 a5  c5
1448 : 0a c5 0c b0 67 a2 02 a0  23
1450 : 02 b5 10 91 0c b5 13 91  41
1458 : 0a ca 88 10 f4 18 a5 11  b4
1460 : 65 10 85 1f a5 12 69 00  a3
1468 : 85 20 18 a5 14 65 13 85  7c
1470 : 21 a5 15 69 00 85 22 a0  cc
1478 : 01 b1 1f 85 1e b1 21 91  e1
1480 : 1f a5 1e 91 21 88 f0 f1  2a
1488 : 18 a5 0a 69 03 85 0a 90  c8
1490 : 02 e6 0b 38 a5 0c e9 03  38
1498 : 85 0c b0 02 c6 0d a5 0d  15
14a0 : c5 0b 90 3b d0 93 a5 0c  cf
14a8 : c5 0a b0 8d 90 31 90 8b  3c
14b0 : d0 8b 90 2b b0 29 a5 1b  f0
14b8 : 48 a5 1c 48 a5 0c 85 1b  ea
14c0 : a5 0d 85 1c 18 90 e7 18  a7
14c8 : a5 19 69 03 85 19 90 01  1a
14d0 : c8 84 1a c4 1c 90 d7 d0  41
14d8 : 26 c5 1b 90 d1 b0 20 a5  28
14e0 : 19 a4 1a c4 0d 90 08 d0  82
14e8 : de c5 0c 90 02 b0 d8 ba  3c
14f0 : e4 fd b0 c2 86 fd e4 fe  41
14f8 : b0 bc a6 fc 9a d0 b1 68  17
1500 : 85 1c 68 85 1b a6 1c e0  77
1508 : 00 d0 bc a6 fc 9a a2 00  a3
1510 : 68 95 0a e8 e0 1a 90 f8  f6
1518 : a9 00 8d 00 ff a9 0f 85  b9
1520 : 1f a9 13 85 20 6c 1f 00  6b
1528 : 95 00 ff 00 ff 00 ff 00  bd
```

Listing 2. »Quicksort 128« bitte mit dem MSE (Seite 158) im C64-Modus eingeben.

```
10 REM ****** QUICKSORT C128 TESTPROGRAMM *******
   **                                              <A99>
15 FAST                                            <M2R>
20 PRINT CHR$(147): COLOR 6,1: COLOR 5,6           <347>
30 PRINT "QUICKSORT-TEST":                         <E97>
40 PRINT "BEFINDET SICH DAS MASCHINENPROGRAMM SCH
   ON IM SPEICHER": INPUT "(J/N)";A$               <T9Q>
50 IF A$ <> "J" THEN BLOAD "PROG 1 OBJ 1300"       <OJ7>
60 PRINT : INPUT "WIEVIELE ELEMENTE ";MAX          <VJN>
70 DIM A$(MAX)                                     <NOQ>
80 GOSUB 200 : REM ERZEUGEN VON ZUFALLSSTRINGS     <675>
90 GOSUB 300 : REM AUSGABE DER STRINGS             <7N1>
100 PRINT : PRINT "JETZT WIRD SORTIERT"            <FTI>
110 T=TI                                           <RRA>
120 SYS DEC("1300")                                <7IA>
130 TT=TI                                          <K2Q>
140 GOSUB 300: REM AUSGABE DER STRINGS             <G31>
150 ZT=(TT-T)/60                                   <9P1>
160 PRINT : PRINT "DAS SORTIEREN VON "MAX" ELEMEN
    TEN DAUERTE "ZT" SEKUNDEN."                    <V5H>
165 SLOW                                           <NEO>
170 END                                            <34E>
200 REM *** ZUFALLSSTRINGS ERZEUGEN ***            <J2C>
210 FOR I=1 TO MAX                                 <3GP>
220 A$(I)=""                                       <FV5>
230 FOR J=1 TO 3                                   <KVI>
240 X=INT(26*RND(X))+65                            <7J5>
250 A$(I)=A$(I)+CHR$(X)                            <SUG>
260 NEXT J                                         <RV5>
270 NEXT I                                         <JI4>
280 RETURN                                         <447>
300 REM *** AUSGABE DES STRINGFELDES ***           <JIE>
310 FOR I=1 TO MAX                                 <7GP>
320 PRINT A$(I),                                   <59V>
330 NEXT I                                         <3K4>
340 RETURN                                         <407>
```

Listing 3. Ein kurzes Demoprogramm in Basic zum Testen des Quicksort-Programms

Sonderzeichen in NLQ-Schrift für Startexter 128

Mathematische oder griechische Sonderzeichen mit dem Startexter 128 auszudrucken, war in eingeschränkter Qualität bisher schon möglich. Mit unserer Erweiterung zum Startexter ist dies in NLQ-Qualität auch auf nicht NLQ-fähigen Druckern wie etwa dem Epson-FX-80 möglich.

Das Programm Startexter von Sybex zeichnet sich durch eine große Anzahl frei definierbarer Sonderzeichen aus, wodurch dieses Programm insbesondere für naturwissenschaftliche Arbeiten interessant wird, bei denen beispielsweise griechische oder mathematische Sonderzeichen benötigt werden. Will man diese Sonderzeichen benutzen, so muß man den Grafik-Druck-Modus wählen, um den selbst definierten Zeichensatz zu Papier zu bringen. Dies bedeutet, daß alle Zeichen, so wie sie auf dem Bildschirm zu sehen sind, in einer Auflösung von 8*8 Punkten ausgedruckt werden. Die Druckqualität ist dabei natürlich nicht überwältigend, von NLQ-Qualität ganz zu schweigen. Abhilfe bringt die hier vorgestellte Erweiterung zu Startexter 128 (Bild 1). Mit ihrer Hilfe stehen zwei neue Befehle zur Verfügung:

<CBM 4> <n>: Einschalten des NLQ-Druck-Modus
<CBM 5> <n>: Ausschalten des NLQ-Druck-Modus

Sie müssen dabei beispielsweise für das Einschalten des NLQ-Modus die Tasten <CBM> und <4> gleichzeitig drücken, anschließend loslassen und <h> drücken.

Ferner ist es möglich, die Tastaturbelegung frei zu definieren. Insgesamt stehen 178 Zeichen zur Verfügung; zirka 14 Sonderzeichen können mehr verwendet werden, als ursprünglich vorgesehen. Dieser NLQ-Modus läßt sich zudem mit jedem anderen Druckmodus kombinieren, so zum Beispiel mit dem gewöhnlichen Grafik-Modus von Startexter oder einer anderen Schrift, die der Drucker bietet.

Wie ist es nun möglich, diese Vorteile auszunutzen? Am einfachsten geht es mit dem »Patching«, wie man es von CP/M her kennt. Tippen Sie hierzu das Programm »PATCH NLQ« (Listing 1) und starten Sie es. Die ASCII/DIN-Taste muß dabei gedrückt sein. Das Programm fordert Sie auf, die Startexter-128-Diskette, oder besser eine Sicherheitskopie, einzulegen, um das Textverarbeitungsprogramm zu laden. Es erscheint am Bildschirm die Meldung »Working«. Danach werden Sie aufgefordert, eine Arbeitsdiskette einzulegen, auf der das Patch-Programm dann eine modifizierte Version des Startexter erstellt. Hierbei wird nur die 80-Zeichen-Version unterstützt. Anschließend kopieren Sie noch Ihre Parameterdatei und Zeichensätze hinzu, und Sie können somit eine fast betriebsbereite NLQ-Startexter-128-Version Ihr eigen nennen, die mit

BOOT "LADER"

gestartet wird. Das »fast« bezieht sich auf den NLQ-Zeichensatz. Bevor das Programm ein Zeichen in NLQ-Qualität ausdrucken kann, muß ihm eine Tabelle mit den entsprechenden Bitmustern zur Verfügung gestellt werden. Dieser NLQ-Zeichensatz umfaßt 256*16*10/8 = 5120 Byte und kann individuell gestaltet sein. Die Nummer, die dieser spezielle NLQ-Grafik-Zeichensatz trägt, muß natürlich auch im Parametermenü eingetragen und anschließend gespeichert werden, so daß dieser Zeichensatz bei jedem Neustart sofort zur Verfügung steht. Dieses Parameter-File erstellt man zweckmäßigerweise mit der unveränderten Startexter-Version und speichert es auf der Diskette mit der »gepatchten« Version.



Bild 1. Diese sehr anschauliche Tastatur-Belegungstabelle wurde mit der neuen Version des Startexter geschrieben

Neue Zeichensätze bequem erstellt

Um das Entwerfen der neuen Zeichensätze so komfortabel wie möglich zu gestalten, finden Sie als Listing 2 das Programm »Charfont«. Dieser Zeicheneditor besteht aus zwei Teilen; einem Basic-Teil, der die Menüs enthält, und einem Maschinenspracheteil (Listing 3) mit den zeitkritischen Routinen. Der Zeicheneditor ist in der Bedienung im wesentlichen kompatibel zum Zeicheneditor »Starfont« und braucht hier daher nicht erklärt zu werden. Wenn Sie die Programmservice-Diskette bestellen, werden Sie darauf einen fertigen Zeichensatz mit mathematischen Symbolen

und entsprechender Tastaturbelegung finden. Des weiteren befindet sich auf der Diskette noch ein Programm namens »Patch F-Key«, das die Tastaturtabelle an den neuen Zeichensatz anpaßt. Die NLQ-Erweiterung wurde für einen Epson-Drucker geschrieben und dürfte mit jedem Epson-kompatiblen Drucker, der über den Linear-Kanal des Interfaces angesteuert wird, zusammenarbeiten.

Nach Einschalten des NLQ-Modus durch <CBM 4> <n> werden alle folgenden Zeichen bis zu dem Befehl <CBM 5> <n> mit einer Auflösung von 16 Zeilen und 10 Spalten ausgegeben. Das entspricht 160 Pixel pro Zeichen im Gegensatz zu 64 im gewöhnlichen Grafikmodus (Faktor 2,5). Bei Verwendung des Dual-Density-Modus (960 Spalten) sind dies 96 Zeichen pro Zeile, was der Elite-Schrift gleichkommt. Die Auflösung von 16 Zeilen wird erreicht, indem jede Zeile in zwei Durchgängen gedruckt und dabei das Papier vor dem zweiten Ausdruck um 1/216 Inch vorgeschoben wird.

Die Steuersequenzen, die hierzu verwendet werden, können vor dem Patching dem jeweiligen Drucker angepaßt werden. Man findet die Steuersequenzen in den Zeilen 2500 und 2530 des Programms »PATCH NLQ« (Listing 1). Die Steuersequenz darf maximal sieben Byte lang sein und muß mit dem Byte $FF abgeschlossen werden.

Zu beachten ist hierbei, daß beim Einschalten des Grafik-Modus auf eine Schriftbreite umgeschaltet werden muß, die der Breite eines NLQ-Zeichens entspricht (für den voreingestellten Grafikmodus ist dies die Elite-Schrift). Für die zweite Druckphase wird anstelle eines »CR« eine entsprechende Anzahl an Backsteps an den Drucker gesandt, um das Umschalten der Druckmodi innerhalb einer Zeile zu ermöglichen.

Um eine individuelle Tastaturbelegung zu ermöglichen, wurden die Tastatur-Dekodierungstabellen in den Bereich von $0c20 bis $0c30 verlegt. Die Adressen im einzelnen:

Normal - Tabelle: $0c20 bis $0c78
Shift - Tabelle: $0c79 bis $0cd1
Commodore-Tabelle: $0cd2 bis $0d2a

Die Tastaturbelegung kann von Startexter aus mit Hilfe des Maschinensprache-Monitors geändert und gespeichert werden. Geben Sie hierzu folgendes ein:

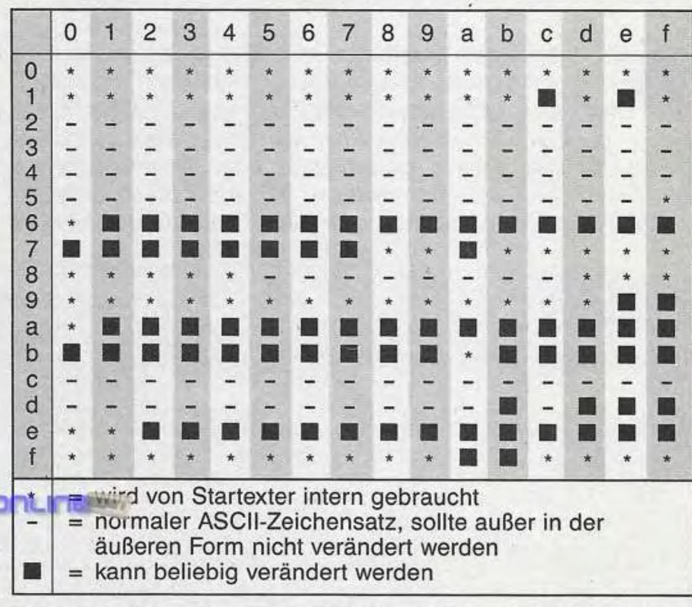
:monitor <control> (Einschalten des Monitors)
m 0c20,0d2a (Auflisten der Tabellen)

Ändern Sie nun die gewünschten Werte gemäß dem C 128-Bedienungshandbuch (Anhang J). Danach können die veränderten Tabellen mit:

s "§:c0" ,8,c00,ed0

gespeichert werden.

Eine Übersicht, welche Zeichen des Zeichensatzes von Startexter nicht benutzt und mit Sonderzeichen belegt werden können (dies sind immerhin 76 Zeichen), finden Sie in Tabelle 1. (Ingolf Kübler/sk)



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
1	*	*	*	*	*	*	*	*	*	*	*	*	■	*	■	*
2	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	*
6	*	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
7	■	■	■	■	■	■	■	■	*	*	■	*	*	*	*	*
8	*	*	*	*	*	–	–	–	–	–	–	–	–	*	*	*
9	*	*	*	*	*	*	*	*	*	*	*	*	*	*	■	■
a	*	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
b	■	■	■	■	■	■	■	■	■	■	*	■	■	■	■	■
c	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
d	–	–	–	–	–	–	–	–	–	–	–	■	–	■	■	■
e	*	*	■	■	■	■	■	■	■	■	■	■	■	■	■	■
f	*	*	*	*	*	*	*	*	*	*	■	■	*	*	*	*

* = wird von Startexter intern gebraucht
– = normaler ASCII-Zeichensatz, sollte außer in der äußeren Form nicht verändert werden
■ = kann beliebig verändert werden

Tabelle 1. Der Zeichensatz von Startexter mit den von Ihnen noch frei definierbaren Zeichen im Überblick

```
1000 REM " £+++++++++++++++++++++++++++++++++++++
     ++++++++++N                              <34C>
1010 REM " K PATCHING STARTEXTER 128{23SPACE}K  <3CD>
1020 REM " K{47SPACE}K                         <3KE>
1030 REM " K (1) FREI DEFINIERBARE TASTATURBELEGU
     NG{8SPACE}K                              <3SF>
1040 REM " K (2) NOCH MEHR SONDERZEICHEN{19SPACE}
     K                                        <248>
1050 REM " K (3) NLQ-SCHRIFT FXR EPSON-KOMPATIBLE
      DRUCKER{2SPACE}K                        <2C9>
1060 REM " K{47SPACE}K                         <2KA>
1070 REM " K COPYRIGHT 1986 BY INGOLF KXBLER{15SP
     ACE}K                                    <2SB>
1080 REM " I+++++++++++++++++++++++++++++++++++++
     ++++++++++@                              <144>
1090 :                                         <5GN>
1100 :                                         <001>
1110 :                                         <0G3>
1120 FAST : GRAPHIC 1 : GRAPHIC 0 : GRAPHIC 5  <EQN>
1130 PRINT "{2HOME,CLR,CTRL+N}" TAB(25) "PATCHING
      STARTEXTER 128"                         <5NA>
1132 WINDOW 10,8,79,24                         <RVR>
1140 :                                         <209>
1150 REM " £+++++++++++++++++++++++++N         <2SB>
1160 REM " K DATEN FXR PATCHING 'B1' K         <248>
1170 REM " I+++++++++++++++++++++++++@         <2C9>
1180 :                                         <40H>
1190 DATA 1D53,4C,A5,0E,-                      <958>
1200 DATA 1DC8                                 <RFR>
1210 DATA 0D,0D,0D,8E,20,20,1F,A6,A6,A6,A6,A6,A6,
     A6,A6,A6                                 <IHH>
1220 DATA A6,9A,A6,A6,A6,A6,A6,A6,05,72,A6,A6,A6,
     1E,A6,A6                                 <NJM>
1230 DATA A6,A6,A6,A6,A6,A6,A6,A6,99,A6,A6,A6,A6,
     A6,A6,A6                                 <6PF>
1240 DATA A6,A6,9E,A6,A6,A6,A6,A6,A6,A6,A6,A6,A6,
     95,A6,A6                                 <GI7>
1250 DATA A6,A6,A6,A6,A6,A6,A6,96,A6,A6,A6,A6,A6,
     A6,A6,A6                                 <B7F>
1260 DATA A6,A6,1C,A6,A6,A6,A6,A6,A6,A6,A6,A6,A6,
     09,09,20                                 <KUJ>
1270 DATA 20,05,66,A6,74,20,54,20,41,20,52,20,20,
     54,20,45                                 <SGP>
1280 DATA 20,58,20,54,20,45,20,52,20,20,20,0E,9B,
     C3,4F,4D                                 <GEU>
1290 DATA 4D,4F,44,4F,52,45,20,31,32,38,20,D0,C3,
     0D,1F,A6                                 <5S9>
1300 DATA A6,A6,A6,A6,A6,A6,A6,A6,A6,9A,A6,A6,A6,
     A6,A6,A6                                 <OUP>
1310 DATA A6,05,A6,A6,A6,8E,67,0E,1E,A6,A6,A6,A6,
     A6,A6,A6                                 <MBT>
1320 DATA A6,A6,99,A6,A6,A6,A6,A6,A6,A6,A6,A6,9E,
     A6,A6,A6                                 <047>
1330 DATA A6,A6,A6,A6,A6,A6,A6,95,A6,A6,A6,A6,A6,
     A6,A6,A6                                 <3AF>
1340 DATA A6,96,A6,A6,A6,A6,A6,A6,A6,A6,A6,A6,1C,
     A6,A6,A6                                 <E7M>
1350 DATA A6,A6,A6,A6,A6,A6,A6,0D,0D,09,09,09,09,
     9D,9D,9B                                 <FSJ>
1360 DATA 56,4F,4E,20,20,D4,4F,4E,49,20,D3,43,48,
     57,41,49                                 <VLP>
1370 DATA 47,45,52,0D,0D,09,09,09,9D,9D,95,4D,4F,
     44,49,46                                 <6GJ>
1380 DATA 49,5A,49,45,52,54,20,56,4F,4E,9E,20,69,
     4E,47,4F                                 <S4M>
1390 DATA 4C,46,20,6B,BD,42,4C,45,52,20,95,31,39,
     38,36,0D,-                               <8C7>
1400 DATA 2039,7F,7F,7F,-                      <OET>
1410 DATA 204C,7F,7F,7F,-                      <E2K>
1480 :                                         <105>
1490 REM " £+++++++++++++++++++++++++N         <3CD>
1500 REM " K DATEN FXR PATCHING 'B3' K         <1K6>
1510 REM " I+++++++++++++++++++++++++@         <1S7>
1520 :                                         <50L>
1530 DATA 757E,20,-                            <Q8H>
1540 DATA 7586,20,-                            <NQ1>
1542 DATA 7004,B5,0E,-                         <OKT>
1550 :                                         <6GR>
1560 REM " £+++++++++++++++++++++++++N         <040>
1570 REM " K DATEN FXR PATCHING 'B4' K         <0C1>
1580 REM " I+++++++++++++++++++++++++@         <3KE>
1590 :                                         <0G3>
1600 DATA A5CA,4C,E5,0D,-                      <V8P>
1610 DATA A659,EA,4C,50,0D,-                   <K5P>
1620 DATA A6A4,09,-                            <CCR>
1630 DATA A6D3,20,75,0D,-                      <LA0>
1640 DATA A6FA,EA,4C,3F,0E,-                   <PUB>
1650 DATA A70B,A2,00,4C,0B,0E,-                <PCK>
1660 DATA A76E,4C,4D,0E,-                      <RCO>
```

```
1670 DATA A677,EA,4C,8D,0D,-                              <TMO>
1680 DATA +                                               <IIO>
1690 :                                                    <3GF>
1700 REM " ++++++++++++++++++++++++++++++++               <0K2>
1710 REM " ! PATCHING 'B1',...,'B4','RB' !                <0S3>
1720 REM " ++++++++++++++++++++++++++++++++               <040>
1730 :                                                    <7GV>
1740 PRINT "LEGEN SIE DIE ORIGINAL STARTEXTER-DIS
     KETTE IN LAUFWERK A"                                 <T8F>
1750 PRINT "UND DRUCKEN SIE EINE TASTE.{CTRL+G}"          <BKA>
1760 GET KEY T$                                           <0OD>
1762 PRINT "{CLR}WORKING..."                              <MCI>
1770 :                                                    <5GN>
1780 BLOAD "B1",ON B0                                     <01V>
1790 BLOAD "B2",ON B0                                     <82V>
1800 BLOAD "B3",ON B0                                     <HFQ>
1810 BLOAD "B4",ON B0                                     <P8Q>
1820 BLOAD "RB",ON B0                                     <RQQ>
1830 :                                                    <8H3>
1832 PRINT "{CLR}LEGEN SIE IHRE ARBEITSDISKETTE I
     N LAUFWERK A"                                        <MMQ>
1834 PRINT "UND DRUCKEN SIE EINE TASTE.{CTRL+G}"          <VIQ>
1836 :                                                    <915>
1840 RESTORE : BANK 0                                     <1MB>
1850 DO                                                   <1C7>
1860 :  READ A$ : IF A$="+" THEN EXIT                     <SQB>
1870 :  A=DEC(A$)                                         <RSO>
1880 :  DO                                                <CPN>
1890 :  :  READ B$ : IF B$="-" THEN EXIT                  <RUC>
1900 :  :  POKE A,DEC(B$) : A=A+1                        <OV7>
1910 :  LOOP                                              <QNP>
1920 LOOP                                                 <089>
1930 :                                                    <9H7>
1932 GET KEY T$                                           <600>
1933 PRINT "{CLR}WORKING..."                              <FVK>
1934 :                                                    <8H3>
1940 BSAVE "C1",ON B0,P(DEC("1D00")) TO P(DEC("21
     B0"))                                                <204>
1950 BSAVE "C2",ON B0,P(DEC("21AA")) TO P(DEC("36
     20"))                                                <CKO>
1960 BSAVE "C3",ON B0,P(DEC("7000")) TO P(DEC("87
     50"))                                                <T9S>
1970 BSAVE "C4",ON B0,P(DEC("A000")) TO P(DEC("AC
     50"))                                                <0VC>
1972 BSAVE "RC",ON B0,P(DEC("0B00")) TO P(DEC("0B
     A0"))                                                <OHC>
1980 :                                                    <C1H>
2050 REM " ++++++++++++++++++++++++++                     <1C5>
2060 REM " ! NEUER LADER ERSTELLEN !                      <1K6>
2070 REM " ++++++++++++++++++++++++++                     <1S7>
2080 :                                                    <30D>
2090 RESTORE 2110                                         <T3F>
2100 :                                                    <60P>
2110 DATA 78,A9,62,8D,18,03,A9,E0,85,2E,A9,FB,85,
     30,A9,00                                             <H6N>
2120 DATA 8D,00,FF,85,FF,8D,00,E0,20,D9,51,A9,0E,
     8D,00,FF                                             <FBL>
2130 DATA A9,05,8D,06,D5,A5,00,09,40,85,00,A5,01,
     09,40,85                                             <8V7>
2140 DATA 01,58,A9,0B,8D,11,D0,A9,01,8D,30,D0,A5,
     BA,AA,A0                                             <LU3>
2150 DATA 01,20,BA,FF,A9,02,A2,6F,A0,1C,20,BD,FF,
     A9,00,20                                             <572>
2160 DATA D5,FF,A9,02,A2,71,A0,1C,20,BD,FF,A9,00,
     20,D5,FF                                             <J92>
2170 DATA EE,72,1C,A9,02,A2,71,A0,1C,20,BD,FF,4C,
     73,1C,52                                             <KFO>
2180 DATA 43,43,30,A9,00,20,D5,FF,20,00,0C,4C,00,
     1D,00,00                                             <POJ>
2190 A=DEC("1C00") : B=DEC("1C80")                        <T5C>
2200 BANK 15                                              <2PD>
2210 FOR I=A TO B-1                                       <2EP>
2220 :  READ A$                                           <LCN>
2230 :  POKE I,DEC(A$)                                    <LFR>
2240 NEXT                                                 <IE0>
2250 BSAVE "@LADER",U8,ON B0,P(A) TO P(B)                 <A0V>
2260 :                                                    <60P>
2270 REM " ++++++++++++++++++++++                         <0S3>
2280 REM " ! FILE 'C0' ERSTELLEN !                        <34C>
2290 REM " ++++++++++++++++++++++                         <3CD>
2300 :                                                    <40H>
2310 BANK 15                                              <IVD>
2320 RESTORE 2340                                         <7BE>
2330 :                                                    <5GN>
2340 DATA AD,C5,0A,09,80,8D,C5,0A,A2,05,BD,1A,0C,
     9D,3E,03                                             <TD8>
2350 DATA CA,10,F7,60,00,00,00,00,00,00,20,0C,79,
     0C,D2,0C                                             <8M5>
2360 :                                                    <70T>
2370 A=DEC("0C00") : B=DEC("0C20")                        <64G>
2380 FOR I=A TO B-1                                       <KEP>
2390 :  READ A$                                           <K8N>
2400 :  POKE I,DEC(A$)                                    <5JO>
2410 NEXT                                                 <MAG>
2420 :                                                    <209>
2430 REM "{2SPACE}TASTATURDECODIERTABELLEN AUS KE
     RNAL KOPIEREN                                        <2SB>
2440 :                                                    <105>
2450 A=DEC("0C20") : B=DEC("0D2C") : C=DEC("FA80"
     )-A                                                  <T9T>
2460 FOR I=A TO B-1                                       <KEP>
2470 :  POKE I,PEEK(C+I)                                  <B4R>
2480 NEXT                                                 <IE0>
2490 :                                                    <7GV>
2500 DATA 1B,4D,1B,4C,FF,FF,FF,FF      : REM " DRUC
     KERCODES FUR ELITE-SCHRIFT EIN                       <5KF>
2510 :                                 : REM " UND
     DUAL-DENSITY-GRAPHIC                                 <28B>
2520 :                                                    <30D>
2530 DATA 1B,4A,01,FF,FF,FF,FF,FF      : REM " DRUC
     KERCODES FUR EINMALIGES LINEFEED                     <51D>
2540 :                                 : REM " VON
     1/216 INCH                                           <78V>
2550 :                                                    <0G3>
2560 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,
     00,00,00                                             <N77>
2570 DATA C9,5A,D0,08,A9,08,8D,02,A7,4C,C7,A6,C9,
     4E,F0,03                                             <MA2>
2580 DATA 4C,5D,A6,8C,46,0D,A9,00,8D,41,0D,A9,01,
     8D,40,0D                                             <QQP>
2590 DATA 8D,47,0D,D0,E1,AC,40,0D,D0,03,4C,F8,A8,
     A2,00,BD                                             <ANN>
2600 DATA 30,0D,C9,FF,F0,06,20,01,AB,E8,D0,F3,60,
     C9,5A,D0                                             <9GQ>
2610 DATA 06,8E,40,0D,4C,B9,A6,C9,4E,F0,03,4C,7B,
     A6,AD,40                                             <LNT>
2620 DATA 0D,F0,EE,AD,41,0D,D0,E9,EA,EA,EA,EA,EA,
     AE,46,0D                                             <2Q3>
2630 DATA AD,47,0D,D0,02,CA,CA,8E,04,8F,A9,01,8D,
     41,0D,A2                                             <ROI>
2640 DATA 08,20,7F,0D,AE,42,0D,F0,08,A9,08,20,01,
     AB,CA,D0                                             <T4C>
2650 DATA F8,20,7F,0D,AD,44,0D,20,01,AB,AD,45,0D,
     20,01,AB                                             <3TE>
2660 DATA A2,01,4C,B9,A6,CD,78,8F,B0,03,4C,DD,A5,
     AD,40,0D                                             <4C1>
2670 DATA D0,0E,A9,00,8D,41,0D,8D,46,0D,8D,47,0D,
     4C,CF,A5                                             <7PN>
2680 DATA AD,41,0D,D0,ED,EA,EA,EA,4C,AD,0D,AD,40,
     0D,D0,08                                             <OE0>
2690 DATA A5,FD,20,01,AB,4C,10,A7,8E,45,0D,A5,FD,
     8D,42,0D                                             <FVL>
2700 DATA 0A,2E,45,0D,0A,2E,45,0D,65,FD,90,03,EE,
     45,0D,0A                                             <OK3>
2710 DATA 2E,45,0D,8D,44,0D,20,01,AB,AD,45,0D,4C,
     12,A7,C9                                             <EGL>
2720 DATA 5A,D0,03,4C,0B,A7,C9,4E,F0,F9,4C,FE,A6,
     AE,40,0D                                             <TBU>
2730 DATA D0,03,4C,04,A8,85,FC,A0,00,84,FD,0A,26,
     FD,0A,26                                             <DD3>
2740 DATA FD,65,FC,90,02,E6,FD,0A,26,FD,0A,26,FD,
     AE,41,0D                                             <R31>
2750 DATA F0,02,69,0A,85,FC,A5,FD,69,EB,85,FD,AE,
     00,FF,8D                                             <VL7>
2760 DATA 01,FF,B1,FC,8E,00,FF,20,01,AB,AE,76,8F,
     F0,0E,AE                                             <80J>
2770 DATA 00,FF,8E,01,FF,B1,FC,8E,00,FF,20,01,AB,
     C8,C0,0A                                             <A2N>
2780 DATA D0,DA,4C,F3,A7,20,D5,FF,A9,4E,8D,71,1C,
     A9,00,20                                             <92N>
2790 DATA D5,FF,4C,56,1D,A2,07,A9,00,9D,40,0D,CA,
     10,FA,4C                                             <8PU>
2792 DATA 26,7F,00,00,00,00,00,00,00,00,00,00,00,
     00,00,00                                             <N1N>
2800 :                                                    <F1T>
2810 A=DEC("0D30") : B=DEC("0ED0")                        <V7L>
2820 FOR I=A TO B-1                                       <CAP>
2830 :  READ A$                                           <NON>
2840 :  POKE I,DEC(A$)                                    <4BV>
2850 NEXT                                                 <64G>
2860 :                                                    <C1H>
2870 BSAVE "C0",U8,ON B0,P(DEC("0C00")) TO P(DEC(
     "0ED0"))                                             <BSE>
2880 :                                                    <B1D>
2890 PRINT "{2HOME,CLR}PATCHING STARTEXTER 128 BE
     ENDET.{2DOWN}"                                       <TAJ>
2900 PRINT "KOPIEREN SIE NUN NOCH IHRE PARAMETERD
     ATEI UND ZEICHENSATZE AUF DIE"                       <G94>
2910 PRINT "ARBEITSDISKETTE UND STARTEN SIE START
     EXTER WIE GEWOHNT.{2DOWN}"                           <F5F>
```

Listing 1. »Patch NLQ« generiert eine Startexter-Version, die Sonderzeichen in NLQ-Qualität druckt. Bitte mit dem Checksummer 128 (Seite 158) eingeben.

```
100 REM " ++++++++++++++++++++++++++++++++++++++++
    ++                                                    <IIA>
110 REM " ! NLQ-ZEICHENSATZ-EDITOR FUR STARTEXTER
     !                                                    <IQB>
120 REM " ! VERSION: DEZEMBER 1986{16SPACE}!              <CRC>
130 REM " !{39SPACE}!                                     <IA9>
140 REM " ! COPYRIGHT 1986 BY INGOLF KUBLER{7SPAC
    E}!                                                   <JIE>
150 REM " ++++++++++++++++++++++++++++++++++++++++
    ++                                                    <JQF>
160 :                                                     <1C5>
1000 FAST : GRAPHIC 1 : GRAPHIC 0 : GRAPHIC 5             <EQN>
1010 PRINT "{2HOME,CLR,BROWN,2SPACE}STARTEXTER{2S
     PACE}CHARFONT{25SPACE}COPYRIGHT 1986 BY INGO
     LF KUBLER{2SPACE}"                                   <HBG>
1020 PRINT "+++++++++++++++++++++++++++++++++++++++
     +++++++++++++++++++++++++++++++++++++++++{
     YELLOW}"                                             <B5R>
1030 :                                                    <0G3>
1040 IF LF=0 THEN BLOAD "CHARFONT.M",ON B0  : REM
      " ASSEMBLERROUTINEN LADEN                           <EP5>
```

Listing 2. »CHARFONT«, der NLQ-Zeicheneditor. Bitte mit dem Checksummer 128 (Seite 158) eingeben.

```
1060 :                                                  <?09>
1070 LF=8                      : REM " DEFAULT-LAU
     FWERK                                              <LA7>
1080 SYS 5417                  : REM " @ DEFINIERE
     N                                                  <VR0>
1090 GOSUB 2400                : REM " ZEICHENSATZ
     IN VDC LADEN                                       <B6F>
1100 GOSUB 3000                : REM " ZEICHENSATZ
     ZEIGEN                                             <56N>
1110 GOSUB 5000                : REM " HELP-SCREEN
     ZEIGEN                                             <B77>
1120 :                                                  <105>
1130 REM " ┌───────────────────────────────────────
     ──┐                                                <3CD>
1140 REM " │ ZEICHEN AUSWAEHLEN / KOMMANDO EINGEBE
     N │                                                <2KA>
1150 REM " └───────────────────────────────────────
     ──┘                                                <2SB>
1160 :                                                  <30D>
1170 XX=0 : YY=0 : ED=0                                 <F5C>
1180 PRINT "{2HOME}" CHR$(27)"S" : WINDOW 62,6,79
     ,24                                                <E9B>
1190 DO                                                 <VFV>
1210 :  IF XX<0 THEN XX=15 : YY=YY-1 : ELSE IF XX
     >15 THEN XX=0 : YY=YY+1                            <NAR>
1220 :  IF YY<0 THEN YY=15              : ELSE IF YY>
     15 THEN YY=0                                       <091>
1230 :  Z=YY*16+XX                                      <G1M>
1240 :  SYS 52625 : SYS 52332,,YY,XX,0 : SYS 5748
     ,Z                                                 <69A>
1250 :  DO                                              <UVV>
1260 :  :  GET KEY B$                                   <34C>
1270 :  :  IF B$="{RIGHT}" THEN XX=XX+1 : EXIT          <DU2>
1280 :  :  IF B$="{LEFT}" THEN XX=XX-1 : EXIT           <IS2>
1290 :  :  IF B$="{DOWN}" THEN YY=YY+1 : EXIT           <HBE>
1300 :  :  IF B$="{UP}" THEN YY=YY-1 : EXIT             <AAE>
1310 :  :  IF B$="M" THEN SYS 5757,Z                    <J7T>
1312 :  :  IF B$="M" THEN SYS 5070,Z : SYS 5320         <4A1>
1320 :  :  IF B$="K" THEN SYS 5639,Z : SYS 4890 :
     EXIT                                               <QH9>
1322 :  :  IF B$="K" THEN SYS 5106,Z : EXIT             <S3P>
1330 :  :  IF B$="L" THEN GOSUB 2400 : EXIT            <264>
1340 :  :  IF B$="S" THEN GOSUB 2200 : EXIT            <190>
1350 :  :  IF B$="V" THEN SYS 5639,Z : SYS 576 :
     SYS 576 : SYS 489 : EXIT                           <421>
1352 :  :  IF B$="V" THEN SYS 5106,Z : SYS 553 :
     SYS 532 : EXIT                                     <IIA>
1360 :  :  IF B$=CHR$(13)  THEN ED=0 : GOSUB 4000
     : EXIT                                             <4GI>
1370 :  :  IF B$=CHR$(141) THEN ED=1 : GOSUB 4000
     : EXIT                                             <NOB>
1390 :  LOOP UNTIL B$="E"                              <FP7>
1400 :  IF B$="E" THEN BEGIN                           <J1H>
1410 :  :  PRINT "{2HOME}" : WINDOW 0,23,79,24,1       <0IS>
1420 :  :  PRINT "{LIG.RED}EDITOR VERLASSEN:{YELL
     OW,3SPACE}SIND SIE SICHER ? ";                     <5VT>
1430 :  :  DO : GET KEY B$ : LOOP UNTIL INSTR("JN
     ",B$)                                              <ALJ>
1440 :  :  IF B$="J" THEN EXIT                         <BEV>
1450 :  :  PRINT "{2HOME}" : WINDOW 62,6,79,24         <GTQ>
1460 :  BEND                                            <RJU>
1470 LOOP                                               <S7P>
1510 BANK 1 : SYS DEC("FC99")                           <NC4>
1520 PRINT "{2HOME,CLR}SYS 65366" CHR$(27)"L"           <2JQ>
1530 POKE 842,19 : POKE 843,13 : POKE 208,2             <PVN>
1540 END                                                <SNM>
2000 :                                                  <70T>
2010 REM " ┌───────────────────────────────┐            <0C1>
2020 REM " │ FILENAME / LAUFWERK EINGEBEN │             <0K2>
2030 REM " └───────────────────────────────┘            <0S3>
2040 :                                                  <50L>
2050 PRINT "{2HOME}" : WINDOW 15,23,79,24,1             <HR3>
2060 PRINT "ZEICHENSATZ ? ";ZS$                         <BMB>
2070 INPUT "{UP,12RIGHT}";ZS$                           <14V>
2080 PRINT "{2HOME}" : WINDOW 45,23,79,24 : PRINT
                                                        <EBK>
2090 PRINT "{HOME}LAUFWERK{4SPACE}? " CHR$(LF+185
     )                                                  <H4V>
2100 DO : INPUT "{HOME,12RIGHT}";U$ : LOOP UNTIL
     INSTR("8A89B9",U$)                                 <045>
2110 IF INSTR("8A8",U$) THEN LF=8 : ELSE LF=9           <MSS>
2120 RETURN                                             <BC7>
2200 :                                                  <50L>
2210 REM " ┌───────┐                                    <1C5>
2220 REM " │ SAVE │                                     <1K6>
2230 REM " └───────┘                                    <1S7>
2240 :                                                  <70T>
2250 PRINT "{2HOME}" CHR$(27)"M" : WINDOW 0,23,79
     ,24,1                                              <6VA>
2260 PRINT "{LIG.RED}SPEICHERN:{YELLOW}"                <7GP>
2270 GOSUB 2000                                         <MN1>
2280 OPEN 8,LF,8,"N"+ZS$+",P,R" : CLOSE  : U=DS         <2AF>
2290 OPEN 8,LF,8,"T"+ZS$+",P,R" : CLOSE 8               <L4G>
2300 IF DS=0 OR U=0 THEN BEGIN                          <3SF>
2310 :  PRINT "{HOME,DOWN}ERSETZEN{4SPACE,CTRL+O}
     ?"                                                 <T3P>
2320 :  DO : GET KEY U$ : LOOP UNTIL INSTR("JN",U
     $)                                                 <UND>
2330 :  IF U$="N" GOTO 2380                             <7KV>
2332 :  PRINT "{HOME,DOWN}" CHR$(27)"Q"                 <URI>
2340 BEND                                               <HUE>
2350 BSAVE ("@N"+ZS$),U(LF) ON B0 ,P(DEC("EB00"))
     TO P(DEC("FEFF"))                                  <QC0>
2360 BSAVE ("@T"+ZS$),U(LF) ON B0 ,P(DEC("3800"))
     TO P(DEC("4000"))                                  <TG6>
2370 :                                                  <7GV>
2380 PRINT "{2HOME}" : WINDOW 0,23,79,24,1 : PRIN
     T "{2HOME}" : WINDOW 62,6,79,24                    <V61>
2390 RETURN                                             <DI7>
2400 :                                                  <30D>
2410 REM " ┌───────┐                                    <2C9>
2420 REM " │ LOAD │                                     <2KA>
2430 REM " └───────┘                                    <2SB>
2440 :                                                  <105>
2450 PRINT "{2HOME}" CHR$(27)"M" : WINDOW 0,23,79
     ,24,1                                              <7FA>
2460 PRINT "{LIG.RED}LADEN:{YELLOW}"                    <RA0>
2470 DO                                                 <NEV>
2480 :  GOSUB 2000                                      <PF2>
2490 :  OPEN 8,LF,8,"T"+ZS$+",P,R" : CLOSE 8            <HO2>
2500 LOOP UNTIL DS=0                                    <5RH>
2510 BLOAD ("T"+ZS$),U(LF) ,ON B0                       <B5V>
2520 :                                                  <30D>
2530 OPEN 8,LF,8,"N"+ZS$+",P,R" : CLOSE 8               <E7S>
2540 IF DS=0 THEN BLOAD ("N"+ZS$),U(LF),ON B0 : E
     LSE SYS DEC("1300")                                <9UQ>
2550 BANK 1 : SYS DEC("131A")   : REM " ZEICHENSAT
     Z NACH VDC KOPIEREN                                <RUL>
2560 :                                                  <105>
2570 PRINT "{2HOME}" : WINDOW 0,23,79,24,1 : PRIN
     T "{2HOME}" : WINDOW 62,6,79,24                    <VA1>
2580 RETURN                                             <B87>
3000 :                                                  <50L>
3010 REM " ┌────────────────────────┐                   <1C5>
3020 REM " │ ZEICHENSATZ ANZEIGEN │                     <1K6>
3030 REM " └────────────────────────┘                   <1S7>
3040 :                                                  <70T>
3050 PRINT "{2HOME,2DOWN}" CHR$(27)"@";                 <P8R>
3060 WINDOW 60,2,79,24                                  <9TL>
3070 PRINT "{LIG.GREEN,2SPACE}ZEICHENSATZ {YELLOW
     }" ZS$                                             <SJ9>
3080 PRINT "{DOWN,BROWN} │0123456789ABCDEF"             <CCS>
3090 PRINT  "┌┼────────────────"                       <DSV>
3100 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F               <A7V>
3110 RESTORE 3100                                       <0BA>
3120 FOR I=0 TO 15                                      <U0N>
3130 :  READ A$                                         <L0N>
3140 :  PRINT "{BROWN}" A$ "{YELLOW} {SYNTH.:142,
     16SPACE,17LEFT,CTRL+N,BROWN}│";                    <QEB>
3150 :  FOR J=0 TO 15                                   <OJH>
3160 :  :  SYS DEC("CDCA"),I*16+J                      <7FE>
3170 :  NEXT                                            <B1G>
3180 :  PRINT "{CTRL+N}"                               <CLA>
3190 NEXT                                               <E9G>
3200 :                                                  <70T>
3210 SYS 5657      : REM " SHAPES LOESCHEN              <UJ0>
3220 PRINT "{CTRL+N,2HOME}" : WINDOW 48,3,79,24         <NOM>
3230 PRINT "{LIG.GREEN} EDIT"                           <62S>
3240 PRINT "{BROWN}┌┬┐ ┌───┐"                           <VV5>
3250 PRINT "│{YELLOW,RVSON}L{RVOFF,BROWN}│ │{YELL
     OW,RVSON}@AB{RVOFF,BROWN}│"                        <F8A>
3260 PRINT "└┴┘ │{YELLOW,RVSON}CDE{RVOFF,BROWN}│"       <GLC>
3270 PRINT "{4SPACE}└───┘{9DOWN}"                       <T4N>
3280 PRINT "{LIG.GREEN} MEMORY:"                        <288>
3290 PRINT "{BROWN,4SPACE}┌───┐"                        <37V>
3300 PRINT "┌┬┐ │{YELLOW,RVSON}FGH{RVOFF,BROWN}│"       <DKK>
3310 PRINT "│{YELLOW,RVSON}M{RVOFF,BROWN}│ │{YELL
     OW,RVSON}IJK{RVOFF,BROWN}│"                        <GMI>
3320 PRINT "└┴┘ └───┘{2HOME,YELLOW}"                    <29E>
3330 :                                                  <7GV>
3340 RETURN                                             <B07>
4000 :                                                  <B1D>
4010 REM " ┌──────────┐                                 <6CP>
4020 REM " │ EDITOR │                                   <6KQ>
4030 REM " └──────────┘                                 <6SR>
4040 :                                                  <915>
4050 SYS 52652 : REM " CURSOR AUS                       <SFS>
4060 PRINT "{2HOME}" : WINDOW 0,2,45,24,1 : PRINT
     "{2HOME}"                                          <FQ0>
4080 :                                                  <F1T>
4090 IF ED THEN BEGIN                                   <F1L>
4100 :  :                                               <JAF>
4110 :  REM " ┌────────────────┐                        <AHB>
4120 :  REM " │ 10X16-EDITOR │                          <BID>
4130 :  REM " └────────────────┘                        <BHF>
4140 :  :                                               <NAV>
4150 :  PRINT "{2HOME,2DOWN,LIG.GREEN}NLQ-EDITOR"       <SIC>
4152 :  PRINT "{BROWN,4SPACE}1{2SPACE}2{2SPACE}3{
     2SPACE}4{2SPACE}5{2SPACE}6{2SPACE}7{2SPACE}8
     {2SPACE}9{2SPACE}10"                               <T15>
4160 :  PRINT "{2SPACE}┌─────────────────────────
     ─────┐"                                           <Q8G>
4170 :  FOR I=1 TO 16                                   <U00>
4180 :  :  PRINT MID$(STR$(I)+"{2SPACE}",2,2) "│{
     YELLOW,30SPACE,BROWN}│" MID$(STR$(I),2)           <807>
4190 :  NEXT                                            <OJG>
4200 :  PRINT "{2SPACE}└─────────────────────────
     ─────┘"                                           <OGR>
4202 :  PRINT "{4SPACE}1{2SPACE}2{2SPACE}3{2SPACE
     }4{2SPACE}5{2SPACE}6{2SPACE}7{2SPACE}8{2SPAC
     E}9{2SPACE}10{2HOME,YELLOW}"                       <4R9>
4210 :  WINDOW 3,5,79,24 : PRINT " "                    <ECP>
4220 :  X=0 : Y=0                                       <DHF>
4230 :  DO                                              <QRF>
4240 :  :  IF X>9  THEN X=0 : Y=Y+1 : ELSE IF X<0
     THEN X=9 : Y=Y-1                                   <5JA>
4250 :  :  IF Y>15 THEN Y=0 : ELSE IF Y<0 THEN Y=
     15                                                 <N11>
4260 :  :  POKE 208,0 : SYS 5751 : SYS 5344             <GKN>
4270 :  :  SYS 52332,,Y,X*3,0 : PRINT "{RVSON,3SP
     ACE,RVOFF}";                                       <AA5>
4290 :  :  GET KEY A$ : A=ASC(A$)                       <JV8>
4300 :  :  IF A=29  THEN X=X+1                          <0AN>
4310 :  :  IF A=157 THEN X=X-1                          <05L>
4320 :  :  IF A=17  THEN Y=Y+1                          <1QP>
4330 :  :  IF A=145 THEN Y=Y-1                          <OLN>
```

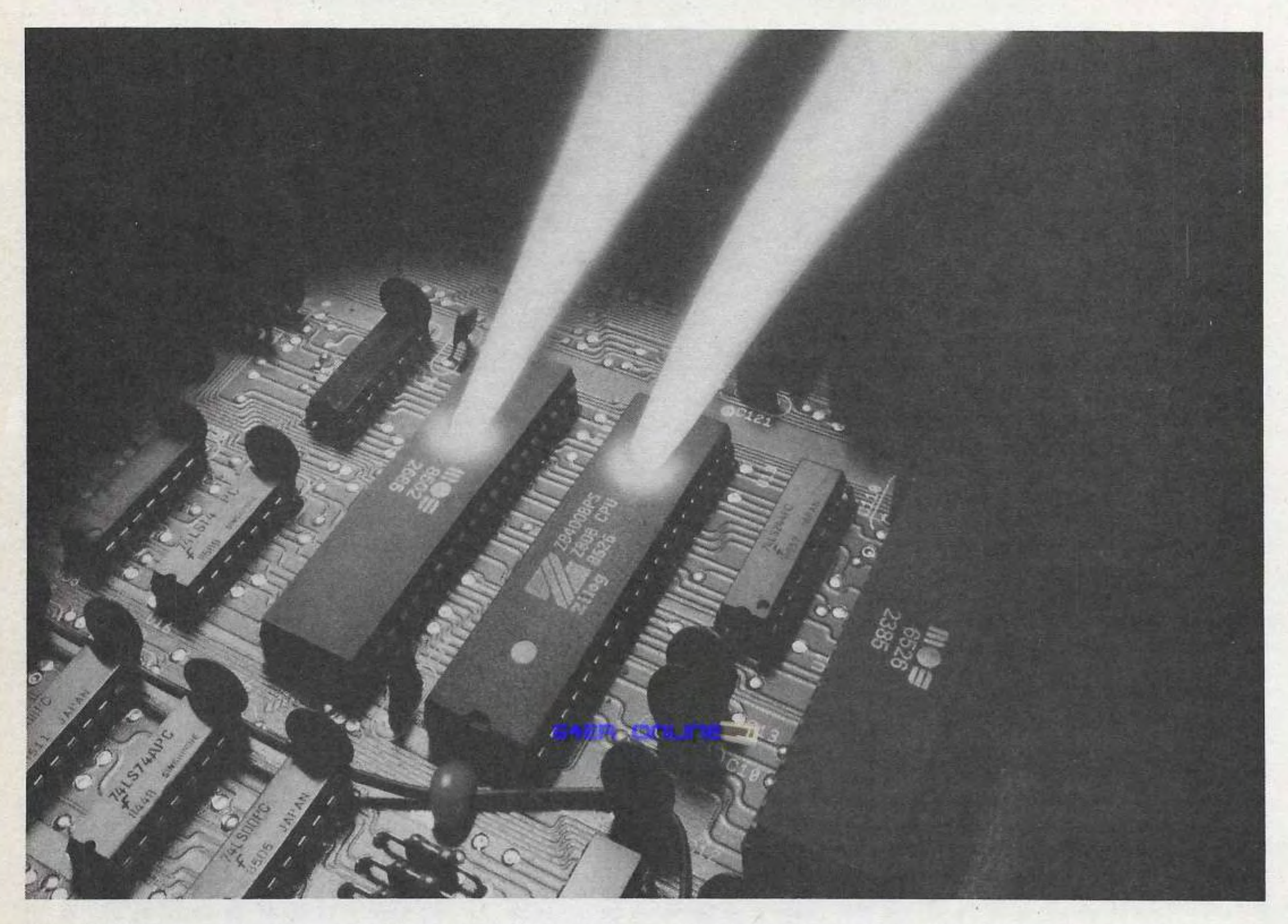
```
4332 :  :  IF A=19  THEN X=0 : Y=0                   <VLU>
4334 :  :  IF A=13  THEN X=0 : Y=Y+1                 <MJO>
4340 :  :  IF A=141 THEN EXIT                        <9QF>
4350 :  :  IF A=83  THEN SYS 5483,,X,Y               <VNK>
4360 :  :  IF A=211 THEN SYS 5483,,X,Y : X=X+1       <57R>
4370 :  :  IF A=68  THEN SYS 5496,,X,Y               <1OF>
4380 :  :  IF A=196 THEN SYS 5496,,X,Y : X=X+1       <R4S>
4390 :  :  IF A=147 THEN SYS 5509                    <C79>
4410 :  LOOP                                         <APP>
4420 :  SYS DEC("13E0"),Z                            <NGB>
4430 BEND : ELSE BEGIN                               <2BN>
4440 :  :                                            <TBN>
4450 :  REM " £++++++++++++N                         <DHN>
4460 :  REM " ¥ 8X8-EDITOR ¥                         <C1H>
4470 :  REM " T++++++++++++@                         <CHJ>
4480 :  :                                            <HA7>
4490 :  PRINT "{2HOME,2DOWN,LIG.GREEN}ZEICHEN-EDI
     TOR"                                            <K3O>
4492 :  PRINT "{BROWN,4SPACE}1 2 3 4 5 6 7 8"        <33E>
4500 :  PRINT "{2SPACE}£+++++++++++++++++N"         <HU2>
4510 :  FOR I=1 TO 8                                 <365>
4520 :  :  PRINT I "{LEFT}¥{YELLOW,16SPACE,BROWN}
     ¥" MID$(STR$(I),2)                              <VAO>
4530 :  NEXT                                         <8HG>
4540 :  PRINT "{2SPACE}T+++++++++++++++++@{2HOME,Y
     ELLOW}"                                         <0DF>
4550 :  WINDOW 3,5,79,24 : PRINT " "                 <BCP>
4560 :  X=0 : Y=0                                    <F1F>
4570 :  DO                                           <EPV>
4580 :  :  IF X>7  THEN X=0 : Y=Y+1 : ELSE IF X<0
      THEN X=7 : Y=Y-1                               <DUQ>
4590 :  :  IF Y>7  THEN Y=0 : ELSE IF Y<0 THEN Y=
     7                                               <IA3>
4600 :  :  SYS 5684 : SYS 5780                      <D24>
4610 :  :  SYS 52332,,Y,X*2,0 : PRINT "{RVSON,2SP
     ACE,RVOFF}";                                    <J9Q>
4630 :  :  GET KEY A$                               <QSF>
4640 :  :  IF A$="{RIGHT}" THEN X=X+1               <BC7>
4650 :  :  IF A$="{LEFT}" THEN X=X-1                <8K8>
4660 :  :  IF A$="{DOWN}" THEN Y=Y+1                <BST>
4670 :  :  IF A$="{UP}" THEN Y=Y-1                  <84I>
4672 :  :  IF A$="{HOME}" THEN X=0 : Y=0            <T1F>
4680 :  :  IF A$=CHR$(13) THEN X=0 : Y=Y+1          <4F3>
4682 :  :  IF A$=CHR$(141) THEN EXIT                <3EC>
4690 :  :  IF A$="S" THEN SYS 5825,,X,Y             <ULB>
4700 :  :  IF A$="S" THEN SYS 5825,,X,Y : X=X+1     <0Q8>
4710 :  :  IF A$="D" THEN SYS 5835,,X,Y             <PCS>
4720 :  :  IF A$="D" THEN SYS 5835,,X,Y : X=X+1     <J4A>
4730 :  :  IF A$="{CLR}" THEN SYS 5845              <MC6>
4750 :  LOOP                                         <QRP>
4760 :  SYS 5621,Z : SYS 4890                        <JL4>
4770 BEND                                            <TNU>
5000 :                                               <915>
5010 REM " £+++++++++++++N                           <7CT>
5020 REM " ¥ HELP-SCREEN ¥                           <7KU>
5030 REM " T+++++++++++++@                           <7SV>
5040 :                                               <B1D>
5050 PRINT "{2HOME,BROWN}" : WINDOW 0,2,45,24,1      <RQP>
5060 PRINT "£+++++++++++++N"                         <G4G>
5070 PRINT "¥{RED} HELP MEN¥: {BROWN}T++++++++++
     ++++++++++++++++++++N"                          <7Q9>
5080 PRINT "¥{GREY3} ZEICHEN MIT CURSOR AUSWFHLEN
     {14SPACE,BROWN}¥"                               <9ME>
5090 PRINT "¥{43SPACE}¥"                             <5PG>
5092 PRINT "¥{GREY3} " CHR$(2)"8X8 ¥10X16¥ BEDEUT
     UNG{19SPACE}" CHR$(130)"{2SPACE,BROWN}¥"        <LMR>
5100 PRINT "¥{GREY3} <CR>¥<CR> ¥ ZEICHEN EDITIERE
     N{13SPACE,BROWN}¥"                              <N0R>
5120 PRINT "¥{GREY3} M{3SPACE}¥ M{3SPACE}¥ AKTUEL
     LES ZEICHEN -> MEMORY{3SPACE,BROWN}¥"           <4DQ>
5130 PRINT "¥{GREY3} K{3SPACE}¥ K{3SPACE}¥ MEMORY
     {2SPACE}-> AKTUELLES ZEICHEN{2SPACE,BROWN}¥"    <6NQ>
5140 PRINT "¥{GREY3} V{3SPACE}¥ V{3SPACE}¥ MEMORY
      <-> AKTUELLES ZEICHEN{2SPACE,BROWN}¥"          <30Q>
5141 PRINT "¥{GREY3,43SPACE,BROWN}¥"                 <4BF>
5142 PRINT "¥{GREY3} L...LOAD{15SPACE}S...SAVE{11
     SPACE,BROWN}¥"                                  <RKQ>
5150 PRINT "¥{GREY3} E...PROGRAMM BEENDEN{22SPACE
     ,BROWN}¥"                                       <3LB>
5160 PRINT "T+++++++++++++++++++++++++++++++++++++
     +++++++@"                                       <BJ6>
5170 PRINT "£+++++++++++++++N"                       <020>
5180 PRINT "¥{RED} HELP EDITOR: {BROWN}T+++++++++
     +++++++++++++++++++++N"                         <KM3>
5200 PRINT "¥{GREY3} PIXEL MIT CURSOR AUSWFHLEN{1
     6SPACE,BROWN}¥"                                 <BAK>
5210 PRINT "¥{GREY3} S........PIXEL SETZEN{15SPAC
     E,SHF.SPACE,5SPACE,BROWN}¥"                     <VGQ>
5220 PRINT "¥{GREY3} D........PIXEL LTSCHEN{20SPA
     CE,BROWN}¥"                                     <HAS>
5230 PRINT "¥{GREY3} RETURN...NFCHSTE ZEILE{20SPA
     CE,BROWN}¥"                                     <OUJ>
5232 PRINT "¥{GREY3} RETURN...EDITIEREN BEENDEN -
     > MEN¥{8SPACE,BROWN}¥"                          <3EE>
5240 PRINT "T+++++++++++++++++++++++++++++++++++++
     +++++++@{2HOME,YELLOW}"                         <7IV>
5250 WINDOW 62,6,79,24                               <E9A>
5260 :                                               <811>
5270 RETURN                                          <PI7>
```

Listing 2. (Schluß)


```
Name : charfont.m           1300 1700
-------------------------------------
1300 : a9 eb 85 65 a9 00 85 64   26
1308 : 8d 01 ff a8 91 64 c8 d0   2c
1310 : fb e6 65 a6 65 e0 ff d0   ab
1318 : f3 60 a9 00 8d 00 ff a2   c4
1320 : 13 a9 00 20 cc cd a9 20   2e
1328 : ca 20 cc cd a9 38 85 65   2c
1330 : a9 00 85 64 a0 00 b1 64   61
1338 : 20 ca cd c8 c0 08 90 f6   c6
1340 : a5 64 69 07 85 64 90 02   14
1348 : e6 65 a9 00 20 ca cd ca   70
1350 : a9 07 20 cc cd a5 65 c9   52
1358 : 40 d0 d9 60 00 00 00 00   83
1360 : 00 00 00 00 00 00 00 00   61
1368 : 00 00 00 00 00 00 00 00   69
1370 : 00 00 00 00 00 00 00 00   71
1378 : 00 00 00 00 00 00 00 00   79
1380 : 00 00 00 00 7c 00 82 82   57
1388 : 82 00 7c 00 00 00 78 84   14
1390 : 00 00 00 84 78 00 00 00   a9
1398 : 85 64 a2 00 86 65 0a 26   00
13a0 : 65 0a 26 65 65 64 90 02   00
13a8 : e6 65 0a 26 65 0a 26 65   92
13b0 : 85 64 a5 65 69 eb 85 65   54
13b8 : 8d 01 ff 60 20 98 13 a0   26
13c0 : 13 b1 64 99 70 13 88 10   da
13c8 : f8 c8 8c 00 ff 60 20 98   fc
13d0 : 13 a0 13 b1 64 99 84 13   7a
13d8 : 88 10 f8 c8 8c 00 ff 60   49
13e0 : 20 98 13 a0 13 b9 70 13   0c
13e8 : 91 64 88 10 f8 c8 8c 00   d7
13f0 : ff 60 20 98 13 a0 13 b9   30
13f8 : 84 13 91 64 88 10 f8 c8   75
1400 : 8c 00 ff 60 a0 13 b9 70   03
1408 : 13 99 5c 13 88 10 f7 60   0b
1410 : a0 13 b9 84 13 99 5c 13   ce
1418 : 88 10 f7 60 5c 66 5c 66   e9
1420 : 5c 66 5c 66 60 6a 60 6a   43
1428 : 60 6a 60 6a 64 6e 64 6e   4b
1430 : 64 6e 64 6e 5c 66 5c 66   e9
1438 : 5c 66 5c 66 60 6a 60 6a   5b
1440 : 60 6a 60 6a 64 6e 64 6e   63
1448 : 64 6e 64 6e 04 04 04 04   43
1450 : 04 04 04 04 04 04 04 04   50
1458 : 04 04 04 04 02 02 02 02   1c
1460 : 02 02 02 02 04 04 04 04   9d
1468 : 04 04 04 04 04 04 04 04   68
1470 : 04 04 04 04 02 02 02 02   34
1478 : 02 02 02 02 20 04 14 a9   02
1480 : 00 8d 00 ff a2 13 20 cc   24
1488 : cd a9 38 ca 20 cc cd a0   72
1490 : 00 b9 1c 14 8d a0 14 b9   98
1498 : 4c 14 aa ca a9 00 ea 1e   75
14a0 : 6e 13 90 04 6a 38 6a 2c   a7
14a8 : 4a 4a ca ea ea 10 f0 20   5a
14b0 : ca cd c8 98 29 07 d0 0b   ca
14b8 : a9 00 20 ca cd a9 07 ca   9f
14c0 : 20 cc cd c0 30 d0 ca 60   47
14c8 : 20 10 14 a9 00 8d 00 ff   97
14d0 : a9 60 a2 13 20 cc cd a9   a7
14d8 : 38 ca 20 cc cd 4c 8f 14   bd
14e0 : 20 04 14 a9 00 8d 00 ff   a9
14e8 : a9 13 20 d2 ff a0 00 b9   f6
14f0 : 19 15 8d fa 14 a2 00 a9   00
14f8 : 20 1e 66 13 90 02 a9 40   64
1500 : 20 d2 ff 20 d2 ff 20 d2   e0
1508 : ff e8 e0 0a d0 e9 a9 0d   12
1510 : 20 d2 ff c8 c0 10 d0 d7   32
1518 : 60 5c 66 5c 66 5c 66 5c   67
1520 : 66 5c 66 5c 66 5c 66 5c   75
1528 : 66 a9 00 8d 00 ff a2 13   c5
1530 : 20 cc cd a9 30 ca 20 cc   d2
1538 : cd a9 aa 20 ca cd ca a9   22
1540 : 07 4c cc cd 86 64 84 65   c2
1548 : 98 a0 00 29 01 f0 02 a0   36
1550 : 0a 98 18 65 64 aa a5 65   56
1558 : 4a a8 60 80 40 20 10 08   74
1560 : 04 02 01 7f bf df ef f7   40
1568 : fb fd fe 20 44 15 b9 5b   b0
1570 : 15 1d 70 13 9d 70 13 60   fd
1578 : 20 44 15 bd 70 13 39 63   03
1580 : 15 9d 70 13 60 a2 13 a9   9d
1588 : 00 9d 70 13 ca 10 fa 60   af
1590 : a2 13 a9 00 9d 84 13 ca   06
1598 : 10 fa 60 a2 13 bd 70 13   99
15a0 : 9d 84 13 ca 10 f7 60 00   e0
15a8 : 00 00 00 10 00 00 00 42   2f
15b0 : 66 5a 42 42 42 42 00 00   52
15b8 : 00 00 00 00 00 00 00 85   c4
15c0 : 64 a9 00 06 64 2a 06 64   32
15c8 : 2a 06 64 2a 69 38 85 65   8d
15d0 : 60 48 20 bf 15 a0 07 b1   2a
15d8 : 64 99 a7 15 88 10 f8 68   53
15e0 : 4c bc 13 ea 20 bf 15 a0   42
15e8 : 07 b1 64 99 af 15 88 10   fa
15f0 : f8 60 4c ce 13 48 20 bf   79
15f8 : 15 a0 07 b9 a7 15 91 64   89
1600 : 88 10 f8 68 4c e0 13 ea   c9
1608 : 20 bf 15 a0 07 b9 af 15   88
1610 : 91 64 88 10 f8 60 4c f2   a1
1618 : 13 a9 00 8d 00 ff a2 13   62
1620 : 20 cc cd a9 38 ca 20 cc   43
1628 : cd a9 00 20 ca cd a9 df   4f
1630 : ca 4c cc cd a9 00 8d 00   de
1638 : ff a9 c0 a2 13 20 cc cd   91
1640 : a9 38 ca 20 cc cd a0 00   7a
1648 : b9 a7 15 20 ca cd c8 c0   de
1650 : 08 d0 f5 60 a9 00 8d 00   1b
1658 : ff a9 d0 a2 13 20 cc cd   b5
1660 : a9 38 ca 20 cc cd a0 00   9a
1668 : b9 af 15 20 ca cd c8 c0   02
1670 : 08 d0 f5 60 20 d1 15 20   8f
1678 : 34 16 4c 7c 14 20 e3 15   56
1680 : 4c 54 16 4c c8 14 a0 07   c3
1688 : b9 a7 15 99 af 15 88 10   73
1690 : f7 60 ea ea a0 00 8c 00   0b
1698 : ff a9 13 20 d2 ff b9 a7   98
16a0 : 15 85 65 a2 08 a9 20 06   80
16a8 : 65 90 02 a9 40 20 d2 ff   5b
16b0 : 20 d2 ff ca d0 ef a9 0d   e0
16b8 : 20 d2 ff c8 c0 08 d0 de   a7
16c0 : 60 b9 a7 15 1d 5b 15 99   be
16c8 : a7 15 60 b9 a7 15 3d 63   28
16d0 : 15 99 a7 15 60 a0 07 a9   b9
16d8 : 00 99 a7 15 88 10 fa 60   e7
16e0 : 00 00 00 00 00 00 00 00   e1
16e8 : 00 00 00 00 00 00 00 00   e9
16f0 : 00 00 00 00 00 00 00 00   f1
16f8 : 00 00 00 00 00 00 00 00   f9
```

Listing 3. »CHARFONT.M«, der Maschinenspracheteil zu »CHARFONT«. Bitte mit dem MSE (Seite 158) im C64-Modus eingeben.

Assembler mit Doppelherz



Darauf haben viele gewartet: Ein Assembler für beide Prozessoren des C128 mit jeder Menge nützlicher Funktionen, der bis zu 58 KByte Quellcode im Speicher verarbeitet.

Mit »Double-Ass« erhalten Sie einen leistungsfähigen 2-Pass-Assembler, mit dem Sie Maschinencode für den Z80- und den 8502-Prozessor (CPU) generieren können. In Tabelle 1 finden Sie einen Überblick über die wichtigsten Eigenschaften von Double-Ass.

Der Assembler ist vollständig in Maschinensprache geschrieben und belegt zirka 9 KByte in Bank 1 des Commodore 128 und nimmt somit keinen Platz des Basic-Programmspeichers ein. Allerdings werden etwa 48 KByte des Variablenspeichers vom Assembler für Labels reserviert. Dieser Speicherbereich wird aber beim Arbeiten mit dem Assembler ohnehin nicht anderweitig benötigt.

Nach der Eingabe von Listing 1 mit dem MSE (bitte beachten Sie die Eingabehinweise auf Seite 158) und Speichern kann der Assembler durch

```
DLOAD "DOUBLE-ASS"
RUN
```

gestartet werden.

Mit einem Boot-Sektor-Programm - Sie finden ein sehr komfortables auf Seite 52 - kann der Boot-Sektor der Diskette so bearbeitet werden, daß der Assembler nach Einschalten des Rechners automatisch gestartet wird.

Nach einem Reset wird der Assembler durch Eingabe von,

```
BANK 1: SYS 5376
```

neu initialisiert.

Der Assembliervorgang selbst wird - wie in Basic - mit »RUN« gestartet, wenn in der ersten Zeile der Befehl »* =« (siehe unten) zusammen mit der Adresse, ab der das Programm generiert werden soll, steht.

Double-Ass fest im Speicher integrieren

Es besteht die Möglichkeit, den Assembler in ein 16K-EPROM (27128) zu brennen und in den freien Steckplatz (U36) des C128 zu stecken. Dies empfiehlt sich besonders bei häufigem Arbeiten mit dem Assembler, aber auch für gelegentliches Benutzen, da so alle anderen Programme völlig unbeeinflußt laufen. Die Kompatibilität bleibt voll erhalten. Um den Assembler auf den Steckplatz U36 zu bannen, benötigt man lediglich ein 16K-EPROM und einen EPROM-Brenner. Das Programm »Double-Ass« kann ohne Änderungen auf das EPROM gebrannt werden.

Schalten Sie bitte anschließend Ihren Computer aus, schrauben ihn auf und stecken Sie das EPROM in den freien Steckplatz U36. Beachten Sie dabei unbedingt die Einbaurichtung (die Kerbe am EPROM muß der Kerbe am Sockel entsprechen). Nach dem Einstecken in U36 des C128 kann der Assembler durch Eingabe von

```
BANK 4: SYS 58449
```

initialisiert werden.

Wo sich der freie Steckplatz U36 auf der Platine befindet und in welchen weiteren Varianten er selektiert werden kann, erfahren Sie in dem Artikel »Der freie Steckplatz« auf Seite 88.

So bedient man den Assembler

In das Betriebssystem wurden zwei zusätzliche Editorfunktionen eingefügt: Die Routinen »FIND« und »MERGE«. »FIND« ermöglicht dem Benutzer, Zeichensequenzen innerhalb eines Quelltextes zu finden. Der Befehl hat folgendes Format:

FIND (Zeichenkette)

Nach dem Befehlswort folgt ohne Leer- oder Anführungszeichen die gesuchte Zeichenfolge.

Mit Hilfe des MERGE-Befehls können verschiedene Quellprogramme aneinandergefügt werden. Er hat folgende Syntax:

MERGE "Dateiname",GN

wobei »GN« die Geräteadresse ist.

Außerdem wurde ein zusätzlicher Basic-Befehl eingebaut, der es ermöglicht, den Assembliervorgang von einem Programm aus zu starten. Er lautet »* =« und wird, gefolgt von der Adresse, ab der das Maschinenprogramm generiert werden soll, eingegeben.

Wie gibt man Source-Code ein?

Ein Maschinenprogramm wird – wie ein Basicprogramm – mit Zeilennummern eingegeben, wobei auch sämtliche Editierfunktionen des Basic-Editors zur Verfügung stehen. Es ist auch möglich, mehrere Befehle durch Doppelpunkt getrennt in eine Zeile zu schreiben, wodurch sich die Übersichtlichkeit der Programme wesentlich erhöhen läßt.

Der Assembler erkennt außerdem, ob der 40-, oder 80-Zeichen-Modus aktiviert ist. Die Ausgabe von Symboltabellen etc. wird entsprechend angepaßt.

Unser Assembler besitzt eine Vielzahl von Pseudo-Opcodes. Das sind Befehle, die dem Assembler zusätzliche Anweisungen für den Assemblierungsablauf, für die Ausgabe von Listings, das Nachladen von Symboltabellen, und so weiter, geben.

Zunächst sollen einige Begriffe, die im Zusammenhang mit den Pseudo-Opcodes Verwendung finden, erklärt werden.

LABEL: Ein Label ist eine Zeichenkette, die für eine Adresse oder einen Wert steht. Das erste Zeichen des Labels muß ein Buchstabe sein. Die folgenden Zeichen können Buchstaben oder Zahlen sein.

Beispiel: VIC001 = $D000

Bei einem Label sind die ersten 40 Zeichen relevant (in Basic sind es nur die ersten beiden). Wichtig: Ein Label darf nicht mit einem reservierten Befehlswort übereinstimmen. Reservierte Befehle sind Pseudo-Opcodes (siehe unten) und Maschinenbefehle des Z80 und 8502.

OPCODE: Die Opcodes sind Befehlswörter für einen Prozessor (z.B. LDA).

PARAMETER: Parameter sind Daten, die der Befehl für seine Ausführung benötigt. Ein Parameter kann auch aus mehreren verknüpften Elementen bestehen. Für die Zusammensetzung eines Parameters lassen sich folgende Operatoren verwenden:

»+« addiert Werte
»–« subtrahiert Werte
»*« multipliziert Werte

- Mischen von Z80- und 8502-(6510-)Maschinencode
- Editierung und Assemblierung von bis zu 58 KByte Quellcode im Speicher
- File-Verkettung möglich (endloses Assemblieren!)
- automatische Anpassung an 40/80-Zeichen-Darstellung
- formatfreie Eingabe über Basic-Editor
- gleichzeitige Verwaltung von bis zu 7000 Labels
- komfortable Ausgabesteuerung durch umfangreiche Pseudocodes
- erweiterter Editor
- auf Wunsch immer resident im Speicher vorhanden

Tabelle 1. Die Leistungsmerkmale unseres Assemblers

»!« logisches OR
»&« logisches AND
»↑« logisches XOR

Die Ausdrücke werden in der Reihenfolge ihres Auftretens abgearbeitet. Falls dies nicht gewünscht wird, so müssen Klammern gesetzt werden.

Die Operanden können folgende Formate haben:

$1000	(Hexadezimal)
4096	(Dezimal)
%10000000	(Binär)
"Z"	(ASCII)
($1000+345*2)	zusammengesetzter Ausdruck

Ist das erste Zeichen des Parameters ein »>« oder ein »<«, so wird entsprechend das High- oder Low-Byte des gesamten nachfolgenden Ausdrucks verarbeitet.

Zu den folgenden Pseudo-Opcodes eine Anmerkung vorab: Die Parameter in eckigen Klammern sind optional, müssen also nicht unbedingt eingegeben werden. Aber nun die Opcodes

=

Format: Label = Parameter

Dieser Pseudo-Opcode weist einem Label einen festen Wert zu. Beispiel:

```
VIC = $D000
```

weist dem Label »VIC« den Wert $D000 zu.

==

Format: Label == Parameter

Dieser Opcode weist einem bereits existierenden Label einen neuen Wert zu.

Beispiel:

```
LAUF = $0001
LAUF == LAUF + 1
```

Z80-Opcodes:

ADC	ADD	AND	BIT	CALL	CCF	CP
CPD	CPDR	CPI	CPIR	CPL	DAA	DEC
DI	DJNZ	EI	EX	EXX	HALT	IM
IN	INC	IND	INDR	INI	JP	JR
LD	LDD	LDDR	LDI	LDIR	NEG	NOP
OR	OTDR	OTIR	OUT	OUTD	OUTI	POP
PUSH	RES	RET	RETN	RL	RLA	RLCA
RLC	RLD	RR	RRA	RRC	RRCA	RRD
RST	SBC	SCF	SET	SLA	SRA	SRL
SUB	XOR					

8502-Opcodes:

ADC	AND	ASL	BIT	BPL	BMI	BVC	BVS
BCC	BCS	BNE	BEQ	CMP	CPX	CPY	DEC
EOR	INC	JMP	JSR	LDA	LDX	LDY	LSR
NOP	ORA	PHA	PHP	PLA	PLP	RTI	RTS
ROL	ROR	SBC	SEC	SED	SEI	TAX	TAY
TSX	TXA	TXS	TYA	STA	STX	STY	

Pseudo-Opcodes:

=	==	*=	.ASC	.BYT	.WOR	.MOD
.OBJ	.LIS	.SYM	.LST	.SST	.END	.FIL
.LF	.IF=	.IF<	.SLO	.FAS	.ERR	

Tabelle 2. Liste aller reservierten Befehle (Z80- und 8502-Opcodes und Pseudo-Opcodes des Assemblers)

*** =**

Format: * = Parameter

Weist dem Program-Counter »PC« einen neuen Wert zu. Achtung! Nach diesem Befehl werden sämtliche durch ».OBJ« geöffnete Dateien geschlossen.

Beispiel:

```
*= $8000
```

gibt dem »PC« die Startadresse, ab der das Programm generiert werden soll, an.

.LIS

Für diesen Befehl gibt es die folgenden drei Formate:

.LIS SCREEN oder .LIS S

Veranlaßt den Assembler, ein formatiertes Listing des Quelltextes auf dem Bildschirm auszugeben.

.LIS NOTHING oder .LIS N

Veranlaßt den Assembler, ohne Ausgabe eines Listings zu assemblieren.

.LIS GA [,SA] [,FN]

Dieser Opcode veranlaßt den Assembler, ein formatiertes Listing auf einem Peripheriegerät (Drucker, Floppy) auszugeben.

GA = Geräteadresse
SA = Sekundäradresse
FN = Dateiname

Dateiname und Sekundäradresse können weggelassen werden. Diese Kürzel gelten auch für weitere Opcodes.

Beispiel:

```
.LIS 4,7
```

gibt ein formatiertes Listing des Quellcodes in Kleinschrift (diese nur auf MPS-Druckern und kompatiblen) auf dem Drucker aus.

.OBJ

Auch für den Befehl ».OBJ« gibt es mehrere Formate:

.OBJ MEMORY [,bank] oder .OBJ M [,bank]
(bank = 0 bis 15)

Ist der Wert »bank« nicht angegeben, wird Bank 0 angenommen. ».OBJ« schreibt den generierten Maschinen-Code im Speicher in die festgelegte RAM-BANK.

.OBJ NOTHING oder .OBJ N

Veranlaßt den Assembler, einen Quelltext lediglich auf Fehler zu überprüfen ohne Maschinencode zu generieren.

.OBJ GA [,SA] [,FN]

Mit diesem Opcode wird der generierte Code auf einem Peripheriegerät ausgegeben.

Beispiel:

```
.OBJ 8,1,"TEST"
```

schreibt den generierten Maschinen-Code unter dem Dateinamen »Test« auf Diskette.

.SYM

Folgende Formate sind möglich:

.SYM SCREEN oder .SYM S

Dieser Opcode veranlaßt den Assembler eine formatierte Symboltabelle auf dem Bildschirm auszugeben.

.SYM GA [,SA] [,FN]

Damit wird eine formatierte Symboltabelle auf ein Peripheriegerät ausgegeben.

Beispiel:

```
.SYM 4, 7
```

gibt eine Symboltabelle auf dem Drucker aus.

.LST (**L**oad **S**ymbol **T**able)

Format: .LST GA, SA, FN

Wobei »SA« normalerweise »0« ist (für »PRG«-Dateien).

Dieser Opcode veranlaßt den Assembler, eine bereits gespeicherte Symboltabelle in den Speicher zu laden und mit zu verarbeiten. Beispiel:

```
.LST 8,0, "TABELLE"
```

lädt die Symboltabelle mit dem Namen »TABELLE«.

.SST (**S**ave **S**ymbol **T**able)

Format: .SST GA, SA, FN

»SA« ist normalerweise »1« (für »PRG«-Dateien).

Dieser Befehl bewirkt, daß die im Speicher befindliche Symboltabelle unter dem Dateinamen »FN« auf Diskette gespeichert wird.

Beispiel:

```
.SST 8,1, "TABELLE"
```

schreibt die Symboltabelle unter dem Namen »TABELLE« auf Diskette.

.MOD

Format: .MOD 1 beziehungsweise .MOD 2

Dieser Opcode schaltet zwischen den beiden Assemblierungsmodi des Assemblers um. ».MOD 1« schaltet auf den Z80-Modus um. Das heißt, nach diesem Opcode darf nur Z80-Code folgen, entsprechend nach ».MOD 2« nur 8502-Code. Die Pseudo-Opcodes bleiben in beiden Modi gleich erhalten.

.FIL

Format: .FIL GA, FN

Dieser Befehl lädt einen weiteren Quelltext nach und assembliert diesen (siehe auch ».END«).

Assemblierung mehrerer Programmteile

.LF

Format: .LF GA, FN

Den Pseudo-Opcode ».LF« leitet »PASS 2« ein (bei mehreren Programmteilen). Besteht ein Quellprogramm aus mehreren Teilen (weil nicht alle gleichzeitig in den Speicher passen), so muß am Ende des ersten Teils der Opcode ».FIL« mit dem Namen des zweiten Teils folgen, welcher dann im ersten Pass assembliert wird. Dies passiert so oft, wie Teile vorhanden sind. Im letzten Teil muß dann am Ende des Quellprogramms der Opcode ».LF GA,FN« stehen, wobei für »FN« der Dateiname des ersten Teils steht, der sodann nachgeladen und im 2. Pass assembliert wird.

.END

Format: .END (optional)

Nach diesem Befehl beginnt der Assembler mit »PASS 2«. Der Befehl wird nur benötigt, wenn kein Assemblerbefehl als letzter Befehl im Listing steht. Es kann also beispielsweise ein Basic-Programm einfach an den Source-Code angehängt werden.

.FAS

Format: .FAS

Dieser Pseudo-Opcode schaltet den Prozessor in den 2 MHz-Modus (nur für die Assemblierung). Hierbei wird der Bildschirm ausgeblendet. Am Ende des Assemblierungsprozesses wird automatisch in den 1 MHz-Modus zurückgeschaltet.

.SLO

Format: .SLO

Schaltet den Prozessor in den 1 MHz-Modus.

.ERR

Format: .ERR

Ist der Prozessor im 2-MHz-Modus, der Bildschirm also ausgeblendet, so können Fehler nicht gesehen werden. Durch diesen Pseudo-Opcode wird, sobald ein Fehler auftritt der Bildschirm eingeblendet. Nach einem beliebigen Tastendruck wird wieder fortgesetzt.

.BYT

Format: .BYT byte, byte, ...

Nach ».BYT« folgen, durch Komma getrennt, beliebig viele Parameter bis zum Ende der Zeile. Hier können also alle Operatoren verwendet werden. »byte« kann Werte von dezimal 0 bis 255 annehmen.

.WOR

Format: .WOR wort, wort, ...

Dieser Befehl ist weitgehend identisch mit »BYT«, nur »wort« kann Werte von dezimal 0-65535 annehmen.

.ASC

Format: .ASC "text"

»text« ist eine Zeichenkette, deren Länge lediglich durch die Länge der Eingabezeile begrenzt ist (160 Zeichen). »text« kann aus allen ASCII-Zeichen bestehen (groß und klein). Die Werte der letzten drei durch die Befehle ».BYT«, ».WOR« und ».ASC« festgelegten Parameter werden direkt im Speicher abgelegt.

.IF<

Format: .IF< Wert 1, Wert 2, Zeilennummer

Dieser Befehl dient zur bedingten Assemblierung. »Wert 1« wird mit dem »Wert 2« verglichen. Ist »Wert 1« kleiner als »Wert 2«, so wird der Assemblierungsvorgang bei der angegebenen Zeilennummer fortgesetzt. Anderfalls wird mit dem ».IF« folgenden Befehl fortgesetzt.

.IF=

Format: .IF= Wert 1, Wert 2, Zeilennummer

Wie ».IF< nur wird hier bei Gleichheit von »Wert 1« und »Wert 2« verzweigt. Mit Hilfe der Befehle ».IF< ».IF=« und »==« können Schleifen aufgebaut werden.

Tips für die Befehlseingabe

Nun noch einige Hinweise zum Eingabeformat: Zwischen Labels und Opcodes (bzw. Pseudo-Opcodes) muß ein Leerzeichen stehen. Ebenso zwischen Opcodes (bzw. Pseudo-Opcodes) und Parametern.

Beispiele:

falsch : * =$8000	richtig : * = $8000
falsch : LDA#1	richtig : LDA #1

Dies dient zusätzlich der Übersichtlichkeit.

In Tabelle 2 finden Sie alle Z80- und 8502-Opcodes sowie die Pseudo-Opcodes des Assemblers im Überblick.

Hinweis zur Zeropage-Adressierung: Der Assembler erkennt selbständig (im 8502-Modus), ob eine Zeropage-Adressierung möglich ist. Sollte dies der Fall sein, benutzt er diese auch. Falls dies nicht gewünscht wird, muß dem gesamten Ausdruck ein Ausrufezeichen »!« vorangehen.

Beispiel:

```
LDA !$80
```

Im Speicher steht dann:

AD 80 00

(»AD« entspricht »LDA« absolut)

So werden beide Prozessor-Codes gemischt

Nach dem Starten des Assemblers befindet sich dieser im 8502-Modus. Um den Assembler auf die Maschinensprache des Z80 bzw. 8502 umzuschalten, existieren die Pseudo-Opcodes ».MOD 1« und ».MOD 2«. Zusätzlich muß dem einen Prozessor aber auch noch mitgeteilt werden, daß er die Kontrolle nun dem anderen Prozessor übergeben muß. Dieses wird nicht durch den Pseudo-Opcode ».MOD« erledigt, da dieser sich nur auf die Steuerung des Assemblers bezieht. Die Umschaltung muß vom Source-Programm selbst erfolgen. Nach der Umschaltung mit ».MOD 1« auf Z80-Code dürfen nur noch Z80-Maschinenbefehle folgen. Umgekehrt nach ».MOD 2« nur noch 8502-Befehle.

Die Vorbereitung zur Übergabe der Kontrolle muß also vorher in 8502-Maschinencode erfolgt sein. Zu dieser Kontrollübergabe nun ein Beispielprogramm:

```
10 *= $8000                        ; PROGRAMM NACH $8000
20 .LIS S                          ; LISTING AUF BILDSCHIRM
30 .OBJ MEMORY                     ; OBJEKTCODE IN SPEICHER
10000 SEI                          ; INTERRUPT VERHINDERN
10010 LDA FF00:PHA                 ; KONFIG.REG. RETTEN
10020 LDA #$C3 :STA $FFEE          ; Z80-BEFEHL JP (JUMP)
10030 LDA #(Z80:STA $FFEF          ; LOW-BYTE
10040 LDA #)Z80:STA $FFF0          ; HIGH-BYTE
10050 LDA #$3E :STA $FF00          ; SETZEN D. KONFIG.REG.
10060 LDA $D505:PHA                ; MCR RETTEN
10070 LDA #$B0 :STA $D505          ; MMU-REGISTER: Z80 EIN
10080 NOP                          ; WARTEN (WICHTIG !!!)
10090 PLA :STA $D505               ; MCR KORRIGIEREN
10100 PLA :STA $FF00               ; KONFIG.REG. KORRIGIEREN
10110 CLI :RTS                     ; INTERRUPT ERLAUBEN, RETURN
11000 Z80 .MOD 1                   ; ES FOLGT Z80-CODE
11010 LD A,$3F                     ; SETZEN DES KONFIGURATIONS-
11020 LD ($FF00),A                 ; REGISTERS
11030 LD A,$51                     ; Z80 -          --
11040 LD ($6000),A                 ;     PROGRAMM  --
11050 JP $FFE0                     ; BETRIEBSSYSTEMROUTINE:
11060                              ; 8502 AKTIVIEREN
```

An diesem Beispiel wollen wir uns einmal genau ansehen, wie das Umschalten gemacht wird. Zunächst ist der 8502-Prozessor aktiv. Nach »STA $D505« in Zeile 10070 wird der 8502 angehalten und der Z80 übernimmt die Kontrolle. Das folgende »NOP« (Zeile 10080) ist notwendig, um gewisse Timing-Probleme der MMU (Memory Managing Unit; dieser Baustein kümmert sich um die Speicherverwaltung) zu überbrücken. Nachdem in Zeile 10100 auch das Konfigurationsregister entsprechend gesetzt wurde, bearbeitet nun die Z80-CPU die Befehle ab Adresse $FFEE (hier hatte der Z80 nach dem Bootversuch von CP/M aufgehört zu arbeiten). Diese Tatsache wird genutzt, um die Übergabe noch im 8502-Modus vorzubereiten. So wurde in der Zeile 10020 ein Z80-Sprungbefehl ($C3 = JP) in die Adresse $FFEE eingegeben. Es folgen das Low-Byte (Zeile 10030) und High-Byte (Zeile 10040) des eigentlichen Z80-Programms. Der Z80-Prozessor springt dann quasi in die Zeile 11010 und arbeitet ab dort weiter.

Um wieder den 8502-Prozessor zu aktivieren, genügt ein Sprung in die Betriebssystem-Routine »$FFE0« (Zeile 11050). Danach fährt der 8502 mit dem »CLI« in Zeile 10110 fort.

Auf diese Weise können Sie die teilweise höhere Arbeitsgeschwindigkeit des Z80 nutzen oder auf Befehle zurückgreifen, die der 8502-Prozessor nicht hat. Beispielsweise kann der Z80 mit einem Befehl einen kompletten Speicherblock verschieben. Auf der Seite 91 finden Sie übrigens eine Einführung in die Z80-Programmierung.

Welche Möglichkeiten sich insgesamt durch das Mischen der beiden Prozessor-Codes dem Programmierer eröffnen, können wir hier nur erahnen. Die Anwendung bleibt Ihrer eigenen Kreativität überlassen.

Interessante Programme können Sie uns gern einschicken. Wir sind schon gespannt darauf. Übrigens, auch das Spiel »Vektors« (Seite 150) wurde mit diesem Assembler geschrieben. (Harald Beine/Arne Jansen/M. Jobst)

```
Name : double-ass          1c01 40a7
----------------------------------------
1c01 : 11 1c 00 00 fe 02 30 3a   56
1c09 : 9e 37 31 38 37 3a a2 00   66
1c11 : 00 00 08 78 a9 3e 8d 00   e5
1c19 : ff a9 05 8d 06 d5 a2 32   de
1c21 : bd 2d 1c 9d 00 0b ca 10   d3
1c29 : f7 4c 93 40 a9 5d 85 fa   c5
1c31 : a9 1c 85 fb a0 00 84 fc   df
1c39 : a9 15 85 fd a2 24 a9 3f   fe
```

Listing 1. »Double-Ass« bitte mit dem MSE (Seite 159) im C64-Modus eingeben.

```
1c41 : 8d 00 ff b1 fa 48 a9 7f   9c
1c49 : 8d 00 ff 68 91 fc c8 d0   a9
1c51 : ed e6 fb e6 fd ca d0 e6   d4
1c59 : 28 4c 00 15 4c 41 35 a2   33
1c61 : 2a bd 85 15 9d 00 02 ca   e5
1c69 : 10 f7 a9 4e 8d 00 ff a9   d5
1c71 : 05 8d 06 d5 a9 40 8d 03   52
1c79 : d5 a9 3f 8d 45 13 ad 30   a8
1c81 : d0 4a 6a 8d 4b 13 a9 38   27
1c89 : a0 f5 8d 2b 13 8d 2d 13   65
1c91 : 8c 2a 13 8c 2c 13 a9 80   8c
1c99 : 8d 3c 13 a5 d7 8d 44 13   df
1ca1 : a9 00 a2 ff 8e 47 13 8e   7f
1ca9 : 48 13 8d 43 13 8d 4a 13   34
1cb1 : 8d 49 13 8d 37 13 8d 41   1e
1cb9 : 13 8d 3f 13 8d 3e 13 8d   f7
1cc1 : 42 13 8d 40 13 a5 3d 8d   67
1cc9 : 35 13 a5 3e 8d 36 13 a9   e3
1cd1 : 1d a2 1c 20 10 29 a9 00   3b
1cd9 : a2 60 85 02 86 03 4c d5   aa
1ce1 : 30 a9 3f 8d 00 ff b1 3d   a9
1ce9 : 48 a9 4e 8d 00 ff 68 60   ad
1cf1 : ae 45 13 8e 00 ff 91 02   23
1cf9 : a2 4e 8e 00 ff 60 a0 00   eb
1d01 : 8c 00 ff 20 32 8e a9 4e   6c
1d09 : 8d 00 ff 60 a9 00 8d 38   e3
1d11 : 13 8d 46 13 ad 00 12 c9   96
1d19 : 3b d0 05 a0 00 4c a4 2e   63
1d21 : a0 00 a9 fe 8d 00 13 a9   84
1d29 : 20 99 00 11 c8 d0 fa a9   8a
1d31 : 20 a2 28 20 93 17 8c 39   47
1d39 : 13 a9 2c a2 2d 20 c6 17   9d
1d41 : 8e 01 13 8e 3d 13 a9 00   fa
1d49 : a2 91 20 c6 17 8e 02 13   a9
1d51 : aa a0 00 8c 13 13 8c 10   f9
1d59 : 13 8c 08 13 8c 0a 13 ad   d8
1d61 : 28 11 d9 ce 19 d0 18 ad   36
1d69 : 29 11 d9 59 1a d0 10 ad   80
1d71 : 2a 11 d9 e4 1a d0 08 ad   da
1d79 : 2b 11 d9 6f 1b f0 07 c8   78
1d81 : c0 8b d0 db a0 ff ad 00   77
1d89 : 13 c9 fe d0 1e 8c 00 13   c7
1d91 : 98 c9 fe 90 23 c8 a9 20   3f
1d99 : 99 00 11 c8 d0 fa a9 20   5b
1da1 : a2 00 8e 38 13 20 93 17   9d
1da9 : 4c d3 15 8c 00 13 98 c9   44
1db1 : fe 90 05 a9 00 4c fd 18   f8
1db9 : a0 00 38 ad 01 13 e9 2d   c8
1dc1 : 8d 01 13 f0 01 c8 ad 02   c3
1dc9 : 13 e9 91 8d 02 13 f0 01   65
1dd1 : c8 8c 03 13 ad 00 13 c9   bd
1dd9 : 76 90 03 4c 50 2a 20 2a   0d
1de1 : 29 ad 3c 13 10 03 4c 98   ce
1de9 : 24 ad 00 13 c9 43 90 05   49
1df1 : a9 00 4c fd 18 ac 03 13   86
1df9 : 88 30 09 88 30 03 20 df   48
1e01 : 18 20 c1 18 ad 00 13 0a   d8
1e09 : a8 b9 35 1c 85 04 b9 36   2b
1e11 : 1c 85 05 a0 00 b1 04 29   35
1e19 : 03 cd 03 13 d0 44 c9 00   7c
1e21 : f0 3d c9 01 f0 1e b1 04   11
1e29 : 29 08 4a 4a 4a cd 0a 13   94
1e31 : d0 30 c8 b1 04 4a 4a 4a   d2
1e39 : 4a 8d 27 13 ae 09 13 20   36
1e41 : 52 19 d0 1e b1 04 29 04   00
1e49 : 4a 4a cd 08 13 d0 13 c8   c2
1e51 : b1 04 29 0f 8d 26 13 ae   e4
1e59 : 07 13 20 52 19 d0 03 4c   f9
1e61 : 2c 17 b1 04 10 05 a9 05   e0
1e69 : 4c fd 18 a2 02 29 03 f0   66
1e71 : 01 e8 b1 04 29 10 f0 01   ac
1e79 : e8 8a 18 65 04 85 04 a9   29
1e81 : 00 65 05 85 05 4c b9 16   ec
1e89 : a0 00 b1 04 48 a0 01 29   f6
1e91 : 03 f0 01 c8 a2 00 b1 04   5f
1e99 : 8d 11 13 68 29 10 f0 07   66
1ea1 : e8 c8 b1 04 8d 12 13 a0   d1
1ea9 : 00 ad 13 13 f0 0d a0 dd   5d
1eb1 : c9 02 d0 02 a0 fd 8c 00   1c
1eb9 : 10 a0 01 ad 11 13 99 00   1f
1ec1 : 10 c8 ad 13 13 f0 01 c8   51
1ec9 : 8a f0 07 ad 12 13 99 00   63
1ed1 : 10 c8 20 04 27 90 03 4c   6a
1ed9 : 06 17 c8 8c 10 13 98 20   6b
1ee1 : cd 2e 20 9c 2f 18 65 02   ae
1ee9 : 85 02 90 02 e6 03 60 85   e7
1ef1 : 04 b9 00 12 f0 2b c9 3b   1a
1ef9 : d0 07 8c 46 13 a9 00 f0   99
1f01 : 20 c8 c9 2c d0 03 ee 38   cf
1f09 : 13 c5 04 f0 0a c9 20 f0   6f
1f11 : e0 9d 00 11 e8 d0 da b9   d6
1f19 : 00 12 c5 04 d0 03 c8 d0   fe
1f21 : f6 60 20 93 17 a9 00 9d   bc
1f29 : 00 11 60 8c 3b 13 20 d2   ce
1f31 : ff c9 0d d0 07 a9 ff 8d   4b
1f39 : 41 13 a9 0d ee 41 13 ac   ae
1f41 : 3b 13 60 48 4a 4a 4a 4a   dc
1f49 : 20 f2 17 68 29 0f c9 0a   fc
1f51 : 90 02 69 06 69 30 4c cf   e6
```

```
1f59 : 17 8d 05 13 a0 00 8c 04   1f
1f61 : 13 88 8c 06 13 bd 00 11   dd
1f69 : c9 28 d0 29 8a 18 6d 05   c9
1f71 : 13 a8 88 b9 00 11 c9 29   34
1f79 : f0 0d a9 01 8d 14 13 68   11
1f81 : 68 ad 14 13 4c fd 18 ee   1a
1f89 : 04 13 a9 02 ce 05 13 ce   c0
1f91 : 05 13 f0 e8 e8 ad 05 13   af
1f99 : c9 01 d0 13 a0 0c bd 00   db
1fa1 : 11 d9 fa 1b f0 03 88 10   2a
1fa9 : f8 8c 06 13 4c 9b 18 c9   61
1fb1 : 02 d0 21 a0 0b bd 00 11   38
1fb9 : d9 07 1c f0 05 88 10 f5   fc
1fc1 : 30 e7 bd 01 11 d9 12 1c   d5
1fc9 : d0 f3 98 18 69 0d 8d 06   fd
1fd1 : 13 4c 9b 18 bd 00 11 c9   a8
1fd9 : 49 d0 1c bd 02 11 c9 2b   6f
1fe1 : d0 15 bd 01 11 c9 58 d0   2e
1fe9 : 05 a0 0b 4c 6e 18 c9 59   0c
1ff1 : d0 05 a0 0c 4c 6e 18 8a   9b
1ff9 : a8 ad 06 13 c9 18 f0 04   85
2001 : c9 19 d0 07 98 18 69 03   62
2009 : a8 d0 05 ad 06 13 10 03   50
2011 : 20 2c 22 ad 05 13 ae 06   35
2019 : 13 ac 04 13 60 a2 2d ad   11
2021 : 01 13 20 fd 17 8d 01 13   7c
2029 : 8e 07 13 8c 08 13 ad 15   8b
2031 : 13 ae 16 13 8d 22 13 8e   d7
2039 : 23 13 60 a2 91 ad 02 13   07
2041 : 20 fd 17 8d 02 13 8e 09   dc
2049 : 13 8c 0a 13 ad 15 13 ae   b5
2051 : 16 13 8d 24 13 8e 25 13   39
2059 : 60 c9 14 08 48 a0 00 a9   81
2061 : 5a 20 cf 17 a9 2e 20 cf   ce
2069 : 17 a5 3c a6 3b 20 1d 02   64
2071 : a9 3a 20 cf 17 68 20 d1   12
2079 : 32 a9 0d 20 cf 17 ad 43   ba
2081 : 13 f0 0d 20 32 30 58 a5   a5
2089 : d0 f0 fc c6 d0 20 10 30   98
2091 : a2 fa 28 90 04 9a 4c 21   55
2099 : 35 9a 20 cd 2e 4c d5 30   da
20a1 : 20 21 21 21 21 20 53 59   d1
20a9 : 4e 54 41 58 0d 00 a0 00   d0
20b1 : 8e 05 13 48 b1 04 29 20   b0
20b9 : d0 1c 68 aa ad 05 13 dd   12
20c1 : ae 19 f0 0f e0 0c f0 09   5e
20c9 : e0 0b d0 07 c9 19 4c 93   02
20d1 : 19 c9 12 4c 93 19 [illegible] aa   d6
20d9 : c9 0e f0 07 ad 05 13 dd   d1
20e1 : be 19 60 b1 04 29 40 f0   e7
20e9 : f3 ad 05 13 c9 0c 60 08   e5
20f1 : a2 01 c9 11 f0 0f c9 18   87
20f9 : f0 0b e8 c9 12 f0 06 c9   36
2101 : 19 f0 02 28 60 8e 13 13   05
2109 : 28 60 00 01 02 03 04 06   d6
2111 : 07 0f 0e 10 13 18 11 ff   5c
2119 : ff ff 02 0a 15 14 09 17   1e
2121 : 16 ff 10 11 12 ff 08 0d   b8
2129 : 0b 00 41 41 41 42 43 43   67
2131 : 43 43 43 43 43 43 44 44   37
2139 : 44 44 45 45 45 48 49 49   e8
2141 : 49 49 49 49 49 4a 4a 4c   53
2149 : 4c 4c 4c 4c 4e 4e 4f 4f   8b
2151 : 4f 4f 4f 4f 50 50 52 52   7b
2159 : 52 52 52 52 52 52 52 52   59
2161 : 52 52 52 52 52 53 53 53   6f
2169 : 53 53 53 53 58 42 42 42   ca
2171 : 42 42 42 42 42 43 43 43   7f
2179 : 45 4a 4a 4c 4c 4c 41 4c   c4
2181 : 52 52 4f 53 53 53 53 42   dc
2189 : 43 43 43 43 44 44 49 49   c5
2191 : 4e 50 50 50 50 52 52 53   ad
2199 : 53 53 54 54 54 54 54 54   17
21a1 : 3d 2a 20 2e 2e 2e 2e 2e   2b
21a9 : 2e 2e 2e 2e 2e 2e 2e 2e   a9
21b1 : 2e 2e 2e 2e 3d 44 44 4e   eb
21b9 : 49 41 43 50 50 50 50 50   e7
21c1 : 50 41 45 49 4a 49 58 58   2d
21c9 : 41 4d 4e 4e 4e 4e 4e 4e   3b
21d1 : 50 52 44 44 44 44 44 45   e6
21d9 : 4f 52 54 54 55 55 55 4f   e5
21e1 : 55 45 45 45 45 4c 4c 4c   53
21e9 : 4c 4c 52 52 52 52 52 53   e2
21f1 : 42 43 45 4c 52 52 55 4f   5b
21f9 : 50 4d 56 56 43 43 4e 45   62
2201 : 4d 50 50 4f 4d 53 44 44   7d
2209 : 44 53 53 4f 4f 52 42 54   ef
2211 : 54 54 52 4c 4c 4c 4c 45   90
2219 : 45 4e 4e 4f 48 48 4c 4c   93
2221 : 54 54 45 45 45 41 41 53   a3
2229 : 58 58 59 20 3d 20 42 57   94
2231 : 41 4d 4f 4c 53 4c 53 45   e6
2239 : 46 4c 49 49 53 46 45 3d   18
2241 : 43 44 44 44 54 4c 46 20 44   42
2249 : 44 49 49 4c 41 43 20 4e   59
2251 : 20 20 58 4c 20 20 43 44   ba
2259 : 44 49 49 20 20 20 44 44   35
2261 : 49 49 47 50 20 44 49 54   1d
2269 : 54 54 50 53 53 54 54 54   38
```

```
2271 : 20 41 43 43 44 20 41 43   3c
2279 : 43 44 54 43 46 54 41 41   ea
2281 : 4c 42 52 4c 49 43 53 43   8f
2289 : 53 45 51 50 58 59 52 50   17
2291 : 52 41 58 59 4c 52 4c 52   f2
2299 : 41 43 41 58 59 4b 43 44   5d
22a1 : 49 56 58 59 58 59 50 41   6b
22a9 : 50 41 50 49 53 43 44 49   ca
22b1 : 58 59 58 41 53 41 20 20   f4
22b9 : 20 59 4f 53 4f 42 49 59   a3
22c1 : 53 53 4e 49 46 46 46 4c   c3
22c9 : 41 52 20 20 20 20 20 4c   5c
22d1 : 20 20 20 52 20 52 20 20   a9
22d9 : 20 20 5a 20 20 20 54 20   39
22e1 : 20 20 20 52 20 52 20 20   b9
22e9 : 20 20 52 20 52 20 20 20   99
22f1 : 52 52 20 44 49 20 48 20   f4
22f9 : 20 49 4e 20 20 41 20 20   22
2301 : 20 20 20 41 20 20 20 20   25
2309 : 20 20 20 20 20 20 20 20   09
2311 : 20 20 20 20 20 20 20 20   11
2319 : 20 20 20 20 20 20 20 20   19
2321 : 20 20 20 20 20 20 20 20   21
2329 : 20 20 20 20 20 20 20 20   29
2331 : 20 20 20 20 20 20 20 20   31
2339 : 20 20 20 20 20 20 20 20   39
2341 : 20 20 20 20 54 52 43 44   eb
2349 : 4a 53 4d 54 54 44 4c 20   f4
2351 : 3d 3c 4f 53 52 20 41 42   9a
2359 : 43 44 45 46 48 4c 5a 50   c9
2361 : 4d 52 49 41 42 44 48 49   4c
2369 : 49 53 4e 4e 50 50 46 43   e0
2371 : 45 4c 58 59 50 5a 43 4f   a1
2379 : 45 50 41 53 53 20 31 20   dc
2381 : 2e 2e 2e 0d 00 50 41 53   22
2389 : 53 20 32 20 2e 2e 2e 0d   a4
2391 : 00 bb 1c e9 1c 22 1d 40   7b
2399 : 1d 64 1d 7f 1d 81 1d 9f   b1
23a1 : 1d a2 1d a5 1d a8 1d ab   ee
23a9 : 1d ad 1d af 1d d9 1d db   a7
23b1 : 1d de 1d e0 1d ec 1d ee   2c
23b9 : 1d f0 1d f2 1d 11 1e 3b   3d
23c1 : 1e 3e 1e 41 1e 44 1e 47   b9
23c9 : 1e 68 1e 77 1e d4 1f d7   47
23d1 : 1f da 1f dd 1f e0 1f e3   1e
23d9 : 1f e5 1f 03 20 06 20 09   d8
23e1 : 20 28 20 2b 20 2e 20 3d   f1
23e9 : 20 4c 20 70 20 8a 20 8d   37
23f1 : 20 9e 20 90 20 92 20 c2   17
23f9 : 20 94 20 e6 20 97 20 0a   9c
2401 : 21 99 20 9b 20 9a 21 9e   03
2409 : 21 9c 21 cc 21 2e 21 52   07
2411 : 21 76 21 0e 22 f0 21 0a   ba
2419 : 90 8e 0a b0 8e 02 00 8f   a1
2421 : 02 10 88 02 20 89 02 30   44
2429 : 8a 02 40 8b 02 50 8c 02   0f
2431 : 60 8d 02 e0 ce 12 89 ed   74
2439 : 4a 12 79 ed 5a 12 99 ed   21
2441 : 6a 92 a9 ed 7a 0a 90 86   64
2449 : 0a b0 86 02 00 87 02 10   f2
2451 : 80 02 20 81 02 30 82 02   ba
2459 : 40 83 02 50 84 02 60 85   ca
2461 : 02 e0 c6 02 89 09 02 79   a1
2469 : 19 02 99 29 02 a9 39 02   65
2471 : 8c 09 02 7c 19 22 99 29   ed
2479 : 22 aa 29 82 ac 39 05 09   46
2481 : a6 05 0b a6 01 00 a7 01   f2
2489 : 01 a0 01 02 a1 01 03 a2   ce
2491 : 01 04 a3 01 05 a4 01 06   23
2499 : a6 81 0e e6 1a 9d cb 46   ab
24a1 : 1a bd cb 46 12 0d cb 47   9d
24a9 : 12 1d cb 40 12 2d cb 41   81
24b1 : 12 3d cb 42 12 4d cb 43   de
24b9 : 12 5d cb 44 92 6d cb 45   43
24c1 : 22 70 dc 22 71 fc 22 72   03
24c9 : d4 22 73 c4 22 74 f4 22   02
24d1 : 75 ec 22 76 e4 22 7c cc   ff
24d9 : a1 07 cd 80 3f 05 09 be   3f
24e1 : 05 0b be 01 00 bf 01 01   40
24e9 : b8 01 02 b9 01 03 ba 01   ef
24f1 : 04 bb 01 05 bc 01 06 bd   1b
24f9 : 81 0e fe 90 ed a9 90 ed   9d
2501 : 89 90 ed a1 90 ed 81 80   01
2509 : 2f 80 27 05 09 35 05 0b   47
2511 : 35 01 00 3d 01 01 05 01   9d
2519 : 02 0d 01 03 15 01 04 1d   e6
2521 : 01 05 25 01 06 2d 01 08   ec
2529 : 0b 01 07 1b 01 09 2b 01   e1
2531 : 0c 2b 81 0a 3b 80 f3 a1   3f
2539 : 0b 10 80 fb 02 9a e3 06   bd
2541 : ca e3 02 27 eb a2 dd 08   cc
2549 : 80 d9 80 76 81 0f 1a 20   de
2551 : ed 78 1a 21 ed 40 1a 22   b3
2559 : ed 48 1a 23 ed 50 1a 24   67
2561 : ed 58 1a 25 ed 60 1a 26   3c
2569 : ed 68 8a e0 db 05 09 34   bb
2571 : 05 0b 34 01 00 3c 01 01   11
2579 : 04 01 02 0c 01 03 14 01   7a
2581 : 04 1c 01 05 24 01 06 2c   2f
```

```
2589 : 01 08 03 01 07 13 01 09   8e
2591 : 23 01 0c 23 81 0a 33 90   f3
2599 : ed aa 90 ed ba 90 ed a2   ea
25a1 : 90 ed b2 21 07 c3 25 08   2c
25a9 : e9 05 0c e9 22 70 da 22   aa
25b1 : 71 fa 22 72 d2 22 73 c2   08
25b9 : 22 74 f2 22 75 ea 22 76   3a
25c1 : e2 a2 7c ca 22 b0 38 22   3a
25c9 : b2 30 22 b3 20 22 bc 28   e9
25d1 : a1 0b 18 06 08 02 06 07   75
25d9 : 12 06 09 77 06 19 70 06   17
25e1 : 29 71 06 39 72 06 49 73   cf
25e9 : 06 59 74 06 69 75 06 e9   a8
25f1 : 36 06 0b 77 06 1b 70 06   e3
25f9 : 2b 71 06 3b 72 06 4b 73   31
2601 : 06 5b 74 06 6b 75 06 eb   e5
2609 : 36 06 0f 32 16 8f ed 43   69
2611 : 16 7f ed 53 06 9f 22 06   bf
2619 : cf 22 16 af ed 73 0a 80   18
2621 : 0a 0a 70 1a 0a 90 7e 0a   c3
2629 : b0 7e 0a f0 3a 02 00 7f   6c
2631 : 02 10 78 02 20 79 02 30   d0
2639 : 7a 02 40 7b 02 50 7c 02   cc
2641 : 60 7d 02 e0 3e 72 ef ed   10
2649 : 57 32 ef ed 5f 0a 91 46   8c
2651 : 0a b1 46 02 01 47 02 11   7a
2659 : 40 02 21 41 02 31 42 02   c2
2661 : 41 43 02 51 44 02 61 45   53
2669 : 02 e1 06 0a 92 4e 0a b2   48
2671 : 4e 02 02 4f 02 12 48 02   01
2679 : 22 49 02 32 4a 02 42 4b   5b
2681 : 02 52 4c 02 62 4d 02 e2   5e
2689 : 0e 0a 93 56 0a b3 56 02   e8
2691 : 03 57 02 13 50 02 23 51   67
2699 : 02 33 52 02 43 53 02 53   87
26a1 : 54 02 63 55 02 e3 16 0a   26
26a9 : 94 5e 0a b4 5e 02 04 5f   4a
26b1 : 02 14 58 02 24 59 02 34   91
26b9 : 5a 02 44 5b 02 54 5c 02   c9
26c1 : 64 5d 02 e4 1e 0a 95 66   46
26c9 : 0a b5 66 02 05 67 02 15   45
26d1 : 60 02 25 61 02 35 62 02   ff
26d9 : 45 63 02 55 64 02 65 65   b2
26e1 : 02 e5 26 0a 96 6e 0a b6   13
26e9 : 6e 02 06 6f 02 16 68 02   3e
26f1 : 26 69 02 36 6a 02 46 6b   ba
26f9 : 02 56 6c 02 66 6d 02 e6   29
2701 : 2e 1a f7 ed 5b 02 f7 11   c0
2709 : 1a f8 ed 4b 02 f8 01 0a   84
2711 : f9 2a 02 f9 21 0a fc 2a   8a
2719 : 02 fc 21 72 fe ed 47 32   11
2721 : fe ed 4f 1a fa ed 7b 02   3e
2729 : 9a f9 02 ca f9 82 fa 31   9c
2731 : 90 ed a8 90 ed b8 90 ed   b7
2739 : a0 90 ed b0 90 ed 44 80   3d
2741 : 00 05 09 b6 05 0b b6 01   63
2749 : 00 b7 01 01 b0 01 02 b1   04
2751 : 01 03 b2 01 04 b3 01 05   8d
2759 : b4 01 06 b5 01 0e f6 90   44
2761 : ed 8b 90 ed b3 16 02 ed   c5
2769 : 79 16 12 ed 41 16 22 ed   59
2771 : 49 16 32 ed 51 16 42 ed   ba
2779 : 59 16 52 ed 61 16 62 ed   5c
2781 : 69 86 0e d3 90 ed ab 90   74
2789 : ed a3 21 0d f1 01 08 c1   fd
2791 : 01 07 d1 01 09 e1 81 0c   68
2799 : e1 21 0d f5 01 08 c5 01   76
27a1 : 07 d5 01 09 e5 81 0c e5   5b
27a9 : 1a 9d cb 86 1a bd cb 86   21
27b1 : 12 0d cb 87 12 1d cb 80   68
27b9 : 12 2d cb 81 12 3d cb 82   c4
27c1 : 12 4d cb 83 12 5d cb 84   21
27c9 : 92 6d cb 85 00 c9 21 00   88
27d1 : d8 21 01 f8 21 02 d0 21   41
27d9 : 03 c0 21 04 f0 21 05 e8   03
27e1 : 21 06 e0 a1 0c c8 90 ed   97
27e9 : 4d 90 ed 45 80 17 80 07   73
27f1 : 90 ed 6f 80 1f 80 0f 90   b7
27f9 : ed 67 15 09 cb 16 15 0b   d8
2801 : cb 16 11 00 cb 17 11 01   d7
2809 : cb 10 11 02 cb 11 11 03   f1
2811 : cb 12 11 04 cb 13 11 05   4e
2819 : cb 14 91 06 cb 15 15 09   df
2821 : cb 06 15 0b cb 06 11 00   c7
2829 : cb 07 11 01 cb 00 11 02   e1
2831 : cb 01 11 03 cb 02 11 04   3b
2839 : cb 03 11 05 cb 04 91 06   9a
2841 : cb 05 15 09 cb 1e 15 0b   0d
2849 : cb 1e 11 00 cb 1f 11 01   64
2851 : cb 18 11 02 cb 19 11 03   7d
2859 : cb 1a 11 04 cb 1b 11 05   da
2861 : cb 1c 91 06 cb 1d 15 09   6b
2869 : cb 0e 15 0b cb 0e 11 00   54
2871 : cb 0f 11 01 cb 08 11 02   6e
2879 : cb 09 11 03 cb 0a 11 04   c7
2881 : cb 0b 11 05 cb 0c 91 06   26
2889 : cb 0d 15 09 cb 26 15 0b   9a
2891 : cb 26 11 00 cb 27 11 01   f0
2899 : cb 20 11 02 cb 21 11 03   09
28a1 : cb 22 11 04 cb 23 11 05   66
28a9 : cb 24 91 06 cb 25 15 09   f8
28b1 : cb 2e 15 0b cb 2e 11 00   ad
28b9 : cb 2f 11 01 cb 28 11 02   c7
28c1 : cb 29 11 03 cb 2a 11 04   20
28c9 : cb 2b 11 05 cb 2c 91 06   7f
28d1 : cb 2d 15 09 cb 3e 15 0b   b2
28d9 : cb 3e 11 00 cb 3f 11 01   05
28e1 : cb 38 11 02 cb 39 11 03   1e
28e9 : cb 3a 11 04 cb 3b 11 05   7b
28f1 : cb 3c 91 06 cb 3d 81 0f   ca
28f9 : 80 37 02 e0 de 0a 90 9e   6f
2901 : 0a b0 9e 02 00 9f 02 10   70
2909 : 98 02 20 99 02 30 9a 02   ee
2911 : 40 9b 02 50 9c 02 60 9d   40
2919 : 12 89 ed 42 12 79 ed 52   fd
2921 : 12 99 ed 62 92 a9 ed 72   db
2929 : 1a 9d cb c6 1a bd cb c6   2a
2931 : 12 0d cb c7 12 1d cb c0   70
2939 : 12 2d cb c1 12 3d cb c2   cd
2941 : 12 4d cb c3 12 5d cb c4   2a
2949 : 92 6d cb c5 05 09 ae 05   9b
2951 : 0b ae 01 00 af 01 01 a8   4c
2959 : 01 02 a9 01 03 aa 01 04   77
2961 : ab 01 05 ac 01 06 ad 81   5e
2969 : 0e ee 05 09 96 05 0b 96   3c
2971 : 01 00 97 01 01 90 01 02   15
2979 : 91 01 03 92 01 04 93 01   1e
2981 : 05 94 01 06 95 81 0e d6   1d
2989 : a9 00 8d 1c 13 8d 19 13   41
2991 : 8d 1a 13 8d 1b 13 b9 00   d3
2999 : 11 f0 25 88 c9 3c f0 04   c7
29a1 : c9 3e d0 04 8d 19 13 c8   bd
29a9 : 20 6a 22 ad 19 13 f0 0f   49
29b1 : a2 00 c9 3e d0 06 ad 16   ae
29b9 : 13 8d 15 13 8e 16 13 60   e1
29c1 : a9 80 8d 1c 13 60 a9 00   6c
29c9 : 8d 17 13 8d 18 13 a9 21   5b
29d1 : 8d 1a 13 c9 00 f0 3d c9   79
29d9 : 2c f0 39 c9 29 f0 3a 20   48
29e1 : cf 22 ad 1a 13 c9 2b f0   7e
29e9 : 19 c9 2d f0 24 c9 2a f0   6b
29f1 : 14 c9 26 f0 13 c9 21 f0   77
29f9 : 12 c9 5e f0 11 a9 06 4c   b5
2a01 : fd 18 4c ea 23 4c 16 24   b0
2a09 : 4c 53 24 4c 68 2[illegible] [illegible] 7d   65
2a11 : 24 4c 00 24 ad 1b 13 d0   82
2a19 : 0d ad 17 13 8d 15 13 ad   4e
2a21 : 18 13 8d 16 13 60 a9 08   d4
2a29 : 4c fd 18 a9 00 8d 15 13   96
2a31 : 8d 16 13 c8 b9 00 11 c9   1b
2a39 : 24 d0 2a c8 b9 00 11 c9   dc
2a41 : 30 90 21 c9 47 b0 1d c9   3d
2a49 : 3a 90 06 c9 41 90 15 69   46
2a51 : 08 a2 04 0a 0a 0a 0a 0a   1a
2a59 : 2e 15 13 2e 16 13 ca d0   63
2a61 : f6 4c df 22 60 c9 25 d0   44
2a69 : 16 c8 b9 00 11 c9 32 b0   db
2a71 : f3 c9 30 90 ef 4a 2e 15   9b
2a79 : 13 2e 16 13 4c 0d 23 c9   d9
2a81 : 30 90 4d c9 3a b0 49 88   e5
2a89 : c8 b9 00 11 38 e9 30 90   05
2a91 : d3 c9 0a b0 cf 8d 1f 13   ed
2a99 : ad 16 13 0a 8d 1e 13 ad   c9
2aa1 : 15 13 0a 2e 16 13 0a 2e   07
2aa9 : 16 13 0a 2e 16 13 0e 15   ee
2ab1 : 13 08 18 6d 15 13 08 18   b6
2ab9 : 6d 1f 13 8d 15 13 ad 16   f9
2ac1 : 13 6d 1e 13 28 69 00 28   93
2ac9 : 69 00 8d 16 13 4c 2c 23   e3
2ad1 : c9 2a d0 0c a5 02 a6 03   70
2ad9 : 8d 15 13 8e 16 13 c8 60   65
2ae1 : c9 22 d0 13 c8 b9 00 11   ce
2ae9 : 8d 15 13 c8 b9 00 11 f0   a0
2af1 : ee c9 22 d0 f6 c8 60 c9   31
2af9 : 28 d0 23 ad 17 13 48 ad   8e
2b01 : 18 13 48 ad 1a 13 48 ee   a4
2b09 : 1b 13 20 6a 22 68 8d 1a   d3
2b11 : 13 68 8d 18 13 68 8d 17   98
2b19 : 13 ce 1b 13 c8 60 c9 41   f6
2b21 : 90 0a c9 5b b0 06 20 98   81
2b29 : 29 90 06 60 a9 07 4c fd   28
2b31 : 18 ad 37 13 30 0b a5 02   46
2b39 : a6 03 8d 15 13 8e 16 13   8b
2b41 : 60 a9 0c 4c fd 18 ad 15   84
2b49 : 13 18 6d 17 13 8d 17 13   c7
2b51 : ad 16 13 6d 18 13 8d 18   fc
2b59 : 13 4c 92 24 ad 17 13 38   0c
2b61 : ed 15 13 8d 17 13 ad 18   40
2b69 : 13 ed 16 13 8d 18 13 4c   d9
2b71 : 92 24 ad 17 13 8d 1d 13   9c
2b79 : ad 18 13 8d 1e 13 a9 00   ca
2b81 : 8d 17 13 8d 18 13 a2 0f   d3
2b89 : 4e 16 13 6e 15 13 90 13   c7
2b91 : ad 1d 13 18 6d 17 13 8d   8c
2b99 : 17 13 ad 1e 13 6d 18 13   8c
2ba1 : 8d 18 13 0e 1d 13 2e 1e   20
2ba9 : 13 ca 10 dc 4c 92 24 ad   06
2bb1 : 18 13 2d 16 13 8d 18 13   85
2bb9 : ad 17 13 2d 15 13 8d 17   ab
2bc1 : 13 4c 92 24 ad 18 13 0d   25
2bc9 : 16 13 8d 18 13 ad 17 13   f0
2bd1 : 0d 15 13 8d 17 13 4c 92   40
2bd9 : 24 ad 18 13 4d 16 13 8d   29
2be1 : 18 13 ad 17 13 4d 15 13   e7
2be9 : 8d 17 13 4c 92 24 b9 00   81
2bf1 : 11 4c 74 22 ad 03 13 d0   6b
2bf9 : 05 a2 00 4c 14 25 c9 02   6f
2c01 : d0 4b a0 91 b9 00 11 aa   06
2c09 : a0 2d b9 00 11 c9 28 d0   50
2c11 : 25 e0 58 f0 13 e0 59 d0   1a
2c19 : 2f ae 3d 13 bd ff 10 c9   01
2c21 : 29 d0 25 a2 01 4c 14 25   5d
2c29 : a0 91 b9 01 11 c9 29 d0   c6
2c31 : 17 a2 02 4c 14 25 e0 58   42
2c39 : d0 05 a2 05 4c 14 25 e0   91
2c41 : 59 d0 05 a2 06 4c 14 25   f5
2c49 : a9 04 4c fd 18 a0 2d b9   76
2c51 : 00 11 c9 23 d0 05 a2 03   76
2c59 : 4c 14 25 c9 28 d0 0f ae   d4
2c61 : 3d 13 bd ff 10 c9 29 d0   2d
2c69 : df a2 04 4c 14 25 a2 07   27
2c71 : 8e 05 13 8a f0 0a a0 2d   d4
2c79 : c9 05 b0 01 c8 20 2c 22   94
2c81 : ad 00 13 c9 3b d0 05 a9   ce
2c89 : 59 8d 00 13 c9 5d 90 0e   f1
2c91 : e9 5d a8 b9 78 26 8d 00   79
2c99 : 10 a0 01 4c a4 2e c9 23   ec
2ca1 : d0 04 a9 ea d0 f0 c9 43   7d
2ca9 : b0 18 a0 05 88 30 0e d9   24
2cb1 : 91 26 d0 f8 98 18 69 5d   53
2cb9 : 8d 00 13 d0 05 a9 00 4c   5b
2cc1 : fd 18 c9 4b b0 43 2c 37   ea
2cc9 : 13 30 05 a0 02 4c a4 2e   bb
2cd1 : 38 e9 43 0a 0a 0a 0a 0a   3d
2cd9 : 09 10 8d 00 10 ad 15 13   37
2ce1 : ae 16 13 38 e9 02 b0 01   da
2ce9 : ca 38 e5 02 8d 01 10 8a   bf
2cf1 : e5 03 d0 0a 2c 01 10 10   f8
2cf9 : d2 a9 0e 4c fd 18 c9 ff   75
2d01 : d0 f7 2c 01 10 30 c4 10   ae
2d09 : f0 c9 4f f0 61 38 e9 4b   e6
2d11 : a8 b9 96 26 48 29 e3 8d   79
2d19 : 00 10 68 29 1c 4a 20 4a   8a
2d21 : 26 a8 b9 ad 26 85 04 b9   d1
2d29 : ae 26 85 05 a0 00 b1 04   c5
2d31 : 29 0f 48 aa ad 05 13 dd   54
2d39 : f1 26 d0 55 b1 04 29 70   dd
2d41 : 4a 4a 0d 00 10 8d 00 10   81
2d49 : a2 03 ac 16 13 68 c9 00   f6
2d51 : d0 02 a2 01 c9 04 90 08   fa
2d59 : ca 20 64 26 c0 00 d0 32   c5
2d61 : ad 15 13 8d 01 10 8c 02   d6
2d69 : 10 8a a8 4c a4 2e a2 4c   51
2d71 : ad 05 13 c9 07 f0 06 c9   42
2d79 : 04 d0 26 a2 6c 8e 00 10   1f
2d81 : ad 15 13 ae 16 13 8d 01   86
2d89 : 10 8e 02 10 a0 03 4c a4   00
2d91 : 2e 68 a0 00 b1 04 30 09   2a
2d99 : e6 04 d0 02 e6 05 4c d2   63
2da1 : 25 a9 05 4c fd 18 48 ad   83
2da9 : 00 13 a2 08 c9 5b f0 0c   30
2db1 : e8 c9 5c f0 07 e8 c9 50   32
2db9 : f0 02 68 60 68 8a 0a 60   94
2dc1 : c9 04 f0 05 c9 08 b0 01   0b
2dc9 : 60 2c 37 13 30 05 a0 00   1d
2dd1 : 8c 16 13 60 00 18 d8 58   0e
2dd9 : b8 ca 88 e8 c8 ea 48 08   4b
2de1 : 68 28 40 60 38 f8 78 aa   fc
2de9 : a8 ba 8a 9a 98 00 02 03   7c
2df1 : 0d 16 c1 e8 c8 41 10 20   ae
2df9 : a1 b6 b8 06 46 26 66 01   15
2e01 : e1 9d 82 80 61 21 30 ce   df
2e09 : ee c3 26 cb 26 d0 26 d3   05
2e11 : 26 d7 26 d9 26 de 26 e3   a1
2e19 : 26 ea 26 ed 26 f0 26 15   a8
2e21 : 31 24 56 63 08 49 f2 20   3d
2e29 : 15 31 56 f2 15 31 84 15   e2
2e31 : 31 56 f2 15 b1 15 31 04   7d
2e39 : 67 f3 15 31 04 56 f2 15   ee
2e41 : 31 56 72 63 08 c9 15 31   2c
2e49 : d7 15 31 d6 81 00 07 05   10
2e51 : 06 03 07 05 06 02 01 29   02
2e59 : 10 4a 4a 4a 4a a8 18 60   75
2e61 : 20 a7 28 a0 00 b1 04 8d   2c
2e69 : 05 13 8c 28 13 8c 29 13   80
2e71 : ac 03 13 f0 e2 88 d0 03   3d
2e79 : 4c a6 27 ae 27 13 29 20   a8
2e81 : d0 11 e0 0d b0 24 e0 0b   79
```

Listing 1. »Double-Ass« (Fortsetzung)

```
2e89 : f0 67 e0 0c d0 74 ee 29  a5
2e91 : 13 d0 6f e0 07 f0 11 e0  02
2e99 : 0b d0 03 4c 76 28 e0 09  95
2ea1 : f0 ec e0 0a f0 e8 d0 5a  cf
2ea9 : a2 0f e0 0f d0 26 a0 01  b0
2eb1 : ad 05 13 29 10 f0 01 c8  e9
2eb9 : ad 13 13 f0 01 c8 ad 24  28
2ec1 : 13 99 00 10 c8 ad 25 13  58
2ec9 : 99 00 10 ee 29 13 ee 29  7d
2ed1 : 13 4c a6 27 ad 25 13 f0  cb
2ed9 : 03 4c 6b 28 a0 01 ad 13  d1
2ee1 : 13 f0 02 a0 03 ad 24 13  55
2ee9 : 99 00 10 ee 29 13 4c a6  0e
2ef1 : 27 ad 25 13 d0 e3 ad 24  c6
2ef9 : 13 8d 02 10 ee 29 13 ee  b8
2f01 : 29 13 ae 26 13 ad 05 13  fd
2f09 : 29 20 d0 14 e0 0d b0 27  80
2f11 : e0 0b d0 03 4c 47 28 e0  6d
2f19 : 0c d0 43 ee 28 13 d0 3e  17
2f21 : e0 07 f0 11 e0 0b d0 03  93
2f29 : 4c 6d 28 e0 09 f0 ec e0  e0
2f31 : 0a f0 e8 d0 29 a2 0f e0  ad
2f39 : 0f d0 26 a0 01 ad 05 13  06
2f41 : 29 10 f0 01 c8 ad 13 13  3b
2f49 : f0 01 c8 ad 22 13 99 00  c3
2f51 : 10 c8 ad 23 13 99 00 10  b3
2f59 : ee 28 13 ee 28 13 4c 58  fb
2f61 : 28 e0 0e d0 1d ad 23 13  89
2f69 : f0 03 4c 6b 28 a0 01 ad  42
2f71 : 13 13 f0 02 a0 03 ad 22  a7
2f79 : 13 99 00 10 ee 28 13 4c  70
2f81 : 58 28 ad 23 13 d0 e3 ad  60
2f89 : 22 13 c9 08 b0 dc a0 01  1f
2f91 : ae 13 13 f0 02 a0 03 0a  f1
2f99 : 0a 0a 19 00 10 99 00 10  dd
2fa1 : 4c 58 28 ad 23 13 d0 c2  6d
2fa9 : ad 22 13 8d 02 10 ee 28  8b
2fb1 : 13 ee 28 13 ad 05 13 29  49
2fb9 : 10 4a 4a 4a 4a 18 6d 29  38
2fc1 : 13 6d 28 13 a8 18 60 38  35
2fc9 : 60 ad 22 13 ae 23 13 4c  d4
2fd1 : 7c 28 ad 24 13 ae 25 13  b3
2fd9 : 38 e9 02 b0 01 ca 38 e5  b0
2fe1 : 02 8d 06 13 8a e5 03 f0  54
2fe9 : 10 c9 ff d0 11 ad 06 13  b4
2ff1 : 10 0c 8d 01 10 a0 01 18  c5
2ff9 : 60 ad 06 13 10 f4 a9 0f  81
3001 : 4c fd 18 ad 00 13 c9 14  f0
3009 : f0 05 c9 3a f0 22 60 ad  32
3011 : 23 13 d0 38 ad 22 13 c9  c5
3019 : 03 b0 31 aa bd d1 28 8d  3c
3021 : 01 10 a9 ed 8d 00 10 a0  ad
3029 : 01 68 68 18 60 46 56 5e  ca
3031 : ad 23 13 d0 17 ad 22 13  dc
3039 : 29 c7 d0 10 ad 22 13 29  06
3041 : 38 09 c7 8d 00 10 a0 00  a4
3049 : 68 68 18 60 a9 09 4c fd  08
3051 : 18 2c 42 13 10 11 48 8a  32
3059 : 48 a2 02 20 c9 ff 68 aa  0a
3061 : 68 20 10 29 20 cc ff 2c  c3
3069 : 40 13 10 10 85 fd 86 fe  99
3071 : a0 00 b1 fd f0 06 20 cf  9d
3079 : 17 c8 d0 f6 a5 fd a6 fe  ea
3081 : 60 a9 0b 4c fd 18 ad 00  5a
3089 : 11 c9 20 d0 02 18 60 a0  45
3091 : 00 a5 02 a6 03 8d 17 13  d8
3099 : 8e 18 13 2c 37 13 30 ed  26
30a1 : 20 98 29 b0 dc ad 2c 13  80
30a9 : ae 2d 13 85 fa 86 fb a0  78
30b1 : 00 ad 17 13 91 fa c8 ad  1f
30b9 : 18 13 91 fa c8 ad 2e 13  f7
30c1 : 91 fa aa 18 69 02 a8 bd  42
30c9 : ff 13 91 fa 88 ca d0 f7  27
30d1 : ad 2e 13 18 69 03 6d 2c  1a
30d9 : 13 8d 2c 13 90 0a ee 2d  90
30e1 : 13 ad 2d 13 c9 b0 f0 02  62
30e9 : 18 60 a9 20 4c fd 18 a9  08
30f1 : 0d 4c fd 18 b9 00 11 a2  cc
30f9 : 00 c9 5b b0 f2 c9 41 90  6e
3101 : ee 9d 00 14 e8 c8 b9 00  fc
3109 : 11 20 13 2a 90 f3 8e 2e  74
3111 : 13 ad 2a 13 ae 2b 13 85  83
3119 : fa 86 fb 8c 2f 13 a5 fb  01
3121 : cd 2d 13 90 07 a5 fa cd  81
3129 : 2c 13 b0 3e a0 02 b1 fa  aa
3131 : cd 2e 13 d0 23 a2 00 c8  cd
3139 : b1 fa dd 00 14 d0 19 c8  9c
3141 : e8 ec 2e 13 90 f2 a0 00  b0
3149 : b1 fa 8d 15 13 c8 b1 fa  b2
3151 : 8d 16 13 ac 2f 13 38 60  71
3159 : a0 02 b1 fa 18 69 03 65  6a
3161 : fa 85 fa 90 02 e6 fb 4c  ce
3169 : c2 29 ac 2f 13 18 60 c9  d8
3171 : 5b b0 0e c9 30 90 0a c9  24
3179 : 41 b0 04 c9 3a b0 02 18  ae
3181 : 60 38 60 ad 2a 13 ae 2b  17
3189 : 13 85 fa 86 fb 4c 43 2a  72
3191 : a0 02 b1 fa 18 69 03 65  a2
3199 : fa 85 fa 90 02 e6 fb a5  b9
31a1 : fb cd 2d 13 90 05 a5 fa  ee
31a9 : cd 2c 13 60 c9 76 d0 22  35
31b1 : 20 c1 18 8d 17 13 8e 18  de
31b9 : 13 ad 00 11 c9 20 d0 05  b0
31c1 : a9 10 4c fd 18 a0 00 20  0c
31c9 : 3f 29 a0 00 4c a4 2e 4c  00
31d1 : 6a 2e c9 8a f0 f9 48 20  56
31d9 : 2a 29 68 c9 77 d0 16 20  82
31e1 : c1 18 85 02 86 03 a9 01  79
31e9 : 20 c3 ff a9 00 8d 3f 13  af
31f1 : a0 00 4c a4 2e c9 78 d0  ee
31f9 : 0f a0 00 ad 00 11 c9 20  fe
3201 : d0 03 4c d5 30 4c a4 2e  75
3209 : c9 79 d0 3c ad 37 13 30  8c
3211 : 07 ac 38 13 c8 4c a4 2e  bd
3219 : 20 aa 2e a0 2d a2 00 8e  33
3221 : 3a 13 20 2c 22 ae 3a 13  19
3229 : ad 16 13 d0 16 ad 15 13  0a
3231 : 9d 00 10 e8 b9 00 11 f0  b1
3239 : d8 c8 c9 2c f0 e1 a9 0a  46
3241 : 4c fd 18 a9 03 4c fd 18  82
3249 : c9 7a d0 3e a9 01 8d 03  2a
3251 : 13 ad 37 13 30 0a ad 38  e6
3259 : 13 0a a8 c8 c8 4c a4 2e  92
3261 : 20 aa 2e a0 2d a2 00 8e  7b
3269 : 3a 13 20 2c 22 ae 3a 13  61
3271 : ad 15 13 9d 00 10 e8 ad  a1
3279 : 16 13 9d 00 10 e8 b9 00  af
3281 : 11 f0 d3 c8 c9 2c f0 df  9a
3289 : d0 b4 c9 7b d0 2d a9 01  b4
3291 : 8d 03 13 20 aa 2e a0 2d  62
3299 : a2 00 b9 00 11 c9 22 d0  33
32a1 : 15 c8 b9 00 11 f0 0a c9  dd
32a9 : 22 f0 06 9d 00 10 e8 d0  3e
32b1 : f0 8a a8 4c a4 2e a9 0e  18
32b9 : 4c fd 18 c9 7c d0 4c 20  03
32c1 : c1 18 e0 00 d0 2a c9 01  4e
32c9 : d0 16 a9 00 8d 3c 13 a9  69
32d1 : 99 a2 2b 2c 37 13 10 03  5e
32d9 : 20 f5 28 a0 00 4c a4 2e  63
32e1 : c9 02 d0 0c a9 80 8d 3c  ae
32e9 : 13 a9 a4 a2 2b 4c 77 2b  98
32f1 : a9 11 4c fd 18 43 50 55  7d
32f9 : 3a 20 5a 2d 38 30 0d 00  b9
3301 : 43 50 55 3a 20 38 35 30  02
3309 : 32 0d 00 c9 7d d0 20 2c  74
3311 : 37 13 10 61 a0 2d b9 00  5c
3319 : 11 c9 4d d0 29 a9 00 ac  b5
3321 : 03 13 c0 01 f0 11 a0 91  3b
3329 : 20 2c 22 ad 16 13 d0 4a  6f
3331 : ad 15 13 c9 04 b0 43 4a  ce
3339 : 6a 6a 09 3f 8d 45 13 a9  a5
3341 : 80 8d 3e 13 d0 2f c9 4e  c4
3349 : f0 1e a9 01 20 3c 30 a2  bd
3351 : 01 20 c9 ff a5 02 20 d2  65
3359 : ff a5 03 20 d2 ff 20 cc  37
3361 : ff a9 80 8d 3f 13 d0 0d  f1
3369 : a9 00 8d 3f 13 8d 3e 13  1a
3371 : a9 01 20 c3 ff a0 00 4c  b9
3379 : a4 2e a9 13 4c fd 18 c9  aa
3381 : 7e d0 35 2c 37 13 10 ed  62
3389 : a0 2d b9 00 11 c9 53 d0  7d
3391 : 07 a9 80 8d 40 13 d0 dd  da
3399 : c9 4e f0 0c a9 02 20 3c  eb
33a1 : 30 a9 80 8d 42 13 d0 cd  13
33a9 : a9 00 8d 42 13 8d 40 13  c3
33b1 : a9 02 20 c3 ff 4c 19 2c  fb
33b9 : c9 7f f0 03 4c fc 2c 2c  94
33c1 : 37 13 10 01 60 a0 2d b9  d9
33c9 : 00 11 c9 53 d0 0e a9 20  93
33d1 : 24 d7 10 02 a9 40 20 9a  78
33d9 : 2c 4c 19 2c a9 04 20 3c  ab
33e1 : 30 a2 04 20 c9 ff a9 40  2b
33e9 : 20 9a 2c 20 cc ff a9 04  e1
33f1 : 20 c3 ff 4c 19 2c 8d 05  af
33f9 : 13 20 27 2a b0 54 a0 02  60
3401 : b1 fa aa c9 09 90 02 a2  75
3409 : 09 c8 b1 fa 20 cf 17 ca  b4
3411 : d0 f7 ad 41 13 29 0f c9  bb
3419 : 09 f0 06 20 c8 2e 4c b6  bd
3421 : 2c a9 3d 20 cf 17 a9 24  1a
3429 : 20 cf 17 a0 01 b1 fa 20  d5
3431 : e7 17 88 b1 fa 20 e7 17  7b
3439 : 20 c8 2e 20 34 2a b0 12  c8
3441 : ad 41 13 29 f0 cd 05 13  31
3449 : 90 b4 a9 0d 20 cf 17 4c  b5
3451 : a2 2c a9 0d 20 cf 17 60  b3
3459 : c9 80 d0 4c 2c 37 13 30  49
3461 : 3d a9 05 20 3c 30 a2 05  92
3469 : 20 c6 ff ad 2c 13 ae 2d  12
3471 : 13 85 fa 86 fb 20 e4 ff  2a
3479 : a0 00 91 fa e6 fa d0 08  77
3481 : e6 fb a5 fb c9 c0 f0 1b  ea
3489 : 24 90 50 e9 20 cc ff a9  02
3491 : 05 20 c3 ff a5 fa a6 fb  5c
3499 : 8d 2c 13 8e 2d 13 a0 00  c1
34a1 : 4c a4 2e a9 20 4c fd 18  8d
34a9 : c9 81 d0 3f 2c 37 13 30  78
34b1 : ed a9 05 20 3c 30 a2 05  92
34b9 : 20 c9 ff ad 2a 13 ae 2b  c0
34c1 : 13 85 fa 86 fb a0 00 b1  4e
34c9 : fa 20 cf 17 e6 fa d0 02  38
34d1 : e6 fb a5 fb cd 2d 13 90  51
34d9 : ee a5 fa cd 2c 13 90 e7  80
34e1 : 20 cc ff a9 05 20 c3 ff  fd
34e9 : 4c 42 2d c9 82 d0 09 ce  4b
34f1 : 03 13 20 cd 2e 4c 26 31  80
34f9 : c9 83 d0 18 a9 80 20 3c  53
3501 : 30 a5 2d a6 2e 38 e9 01  72
3509 : b0 01 ca 85 3d 86 3e a0  df
3511 : 00 4c a4 2e c9 84 d0 10  4a
3519 : 2c 37 13 30 05 a9 80 20  8c
3521 : 3c 30 20 cd 2e 4c 26 31  77
3529 : c9 85 d0 4d a0 2d 20 cb  1e
3531 : 30 85 fa 86 fb a0 91 20  fe
3539 : cb 30 c5 fa d0 2e e4 fb  f7
3541 : d0 2a b9 00 11 c9 2c d0  46
3549 : 2b c8 20 cb 30 85 16 86  ee
3551 : 17 a9 4c 8d 00 ff 20 64  4b
3559 : 50 a9 4e 8d 00 ff 90 39  78
3561 : a5 61 e9 01 85 3d a5 62  ef
3569 : e9 00 85 3e ee 03 13 a0  10
3571 : 00 4c a4 2e a9 05 4c fd  76
3579 : 18 c9 86 d0 21 a0 2d 20  3e
3581 : cb 30 85 fa 86 fb a0 91  13
3589 : 20 cb 30 c5 fa f0 02 b0  f4
3591 : b1 e4 fb f0 d7 b0 ab 90  a4
3599 : d3 a9 1f 4c fd 18 c9 87  69
35a1 : d0 0d a9 00 8d 44 13 20  ea
35a9 : 10 30 a0 00 4c a4 2e c9  30
35b1 : 88 d0 04 a9 80 d0 ed c9  b1
35b9 : 89 d0 0b 20 c1 18 8d 43  0b
35c1 : 13 a0 00 4c a4 2e c9 8a  a6
35c9 : d0 36 20 c1 18 8d 17 13  65
35d1 : 8e 18 13 ad 00 11 c9 20  d6
35d9 : d0 05 a9 10 4c fd 18 a0  ef
35e1 : 00 20 98 29 90 12 a0 00  59
35e9 : ad 17 13 91 fa c8 ad 18  f6
35f1 : 13 91 fa a0 00 4c a4 2e  f1
35f9 : a0 00 20 49 29 4c 97 2e  7a
3601 : 8c 10 13 4c 82 17 ac 39  ea
3609 : 13 a2 2d b9 00 12 9d 00  f7
3611 : 11 e8 c8 c9 00 d0 f4 60  1d
3619 : ec 41 13 90 06 20 c8 2e  5e
3621 : 4c bc 2e 60 a9 20 4c cf  cf
3629 : 17 2c 37 13 30 01 60 48  a4
3631 : ad 40 13 f0 03 20 ee 2e  2b
3639 : ad 42 13 f0 0b a2 02 20  f8
3641 : c9 ff 20 ee 2e 20 cc ff  07
3649 : 68 60 ad 00 12 c9 3b d0  4b
3651 : 03 4c 88 2f a5 03 20 e7  45
3659 : 17 a5 02 20 e7 17 a9 3a  1a
3661 : 20 cf 17 20 c8 2e 20 c8  43
3669 : 2e ac 10 13 f0 15 c0 04  17
3671 : 90 02 a0 04 a2 00 bd 00  cc
3679 : 10 20 e7 17 20 c8 2e e8  49
3681 : 88 d0 f3 a2 15 20 bc 2e  64
3689 : a0 09 a2 00 bd 00 11 20  b7
3691 : cf 17 e8 88 d0 f6 20 c8  0e
3699 : 2e 20 c8 2e a0 04 a2 00  84
36a1 : bd 28 11 20 cf 17 e8 88  25
36a9 : d0 f6 20 c8 2e ac 03 13  90
36b1 : f0 1b c0 01 f0 12 a2 2d  04
36b9 : 20 7c 2f a9 2c 20 cf 17  4a
36c1 : a2 91 20 7c 2f 4c 71 2f  3d
36c9 : a2 2d 20 7c 2f ac 46 13  31
36d1 : d0 1a a9 0d 20 cf 17 60  58
36d9 : bd 00 11 f0 06 20 cf 17  c8
36e1 : e8 d0 f5 60 a0 06 20 c8  07
36e9 : 2e 88 d0 fa b9 00 12 f0  b4
36f1 : e1 20 cf 17 c8 4c 90 2f  49
36f9 : 2c 37 13 30 01 60 48 ad  1b
3701 : 3e 13 f0 0f ac 10 13 f0  60
3709 : 0a 88 b9 00 10 20 0f 02  08
3711 : 88 10 f7 ad 3f 13 f0 1b  db
3719 : a2 01 20 c9 ff ac 10 13  49
3721 : f0 0e a2 00 bd 00 10 20  1d
3729 : d2 ff e8 ec 10 13 d0 f4  99
3731 : 20 cc ff a5 03 cd 48 13  52
3739 : 90 09 d0 11 a5 02 cd 47  d4
3741 : 13 b0 0a a5 02 8d 47 13  b3
3749 : a5 03 8d 48 13 a5 02 18  73
3751 : 6d 10 13 a6 03 90 01 e8  eb
3759 : ec 4a 13 90 0d d0 05 cd  48
3761 : 49 13 90 06 8d 49 13 8e  a5
3769 : 4a 13 68 60 ad 44 13 0a  c0
3771 : 2a 2c 11 d0 10 fb 8d 30  87
3779 : d0 0a 0a 0a 0a 49 10 8d  59
3781 : 05 13 ad 11 d0 29 6f 0d  cb
3789 : 05 13 8d 11 d0 60 a9 00  54
3791 : f0 dd 2c 37 13 30 01 60  d9
3799 : 48 20 c3 ff 20 aa 2e a0  33
37a1 : 2d 20 2c 22 ad 16 13 d0  a7
37a9 : 66 ad 15 13 85 ba 20 bd  b8
37b1 : 30 b0 3f 68 48 30 12 20  a5
```

```
37b9 : 2c 22 ad 16 13 d0 50 ad   79
37c1 : 15 13 85 b9 20 bd 30 b0   0a
37c9 : 29 b9 00 11 c9 22 d0 3f   60
37d1 : c8 84 bb 88 a2 00 c8 b9   9c
37d9 : 00 11 9d e0 03 e8 f0 04   28
37e1 : c9 22 d0 f2 98 38 e5 bb   a8
37e9 : 85 b7 a9 e0 85 bb a9 03   b3
37f1 : 85 bc 68 10 12 a9 03 85   76
37f9 : b8 a9 00 85 b9 a6 2d a4   05
3801 : 2e 20 d5 ff b0 0e 60 85   bd
3809 : b8 20 c0 ff b0 06 60 a9   11
3811 : 12 4c fd 18 a9 1e 4c fd   84
3819 : 18 b9 00 11 f0 07 c8 c9   2e
3821 : 2c d0 ec 18 60 38 60 20   7d
3829 : 2c 22 ad 15 13 ae 16 13   99
3831 : 60 a2 fa 9a a5 91 c9 7f   01
3839 : d0 0a a9 a3 a2 31 20 10   41
3841 : 29 4c 66 31 20 fc 31 20   3f
3849 : 19 32 a9 00 8d 33 13 8d   c0
3851 : 34 13 a0 00 20 00 02 d0   e3
3859 : 16 c8 c8 20 00 02 f0 22   22
3861 : c8 20 00 02 85 3b c8 20   0f
3869 : 00 02 85 3c 20 19 32 a2   2c
3871 : 00 20 25 32 9d 00 12 e8   05
3879 : c9 00 d0 f5 20 b0 15 4c   aa
3881 : d5 30 ad 37 13 49 80 8d   59
3889 : 37 13 f0 36 ad 35 13 ae   7b
3891 : 36 13 85 3d 86 3e a9 00   5b
3899 : a2 60 85 02 86 03 a9 29   86
38a1 : a2 1c 20 10 29 a9 99 a2   e7
38a9 : 2b 20 f5 28 a9 80 8d 3c   b4
38b1 : 13 a9 00 8d 3e 13 8d 3f   7c
38b9 : 13 8d 40 13 8d 42 13 4c   d5
38c1 : d5 30 a9 01 20 c3 ff a9   ac
38c9 : 02 20 c3 ff a9 c9 a2 31   a2
38d1 : 20 10 29 ad 48 13 20 e7   67
38d9 : 17 ad 47 13 20 e7 17 a9   ec
38e1 : e2 a2 31 20 10 29 ad 4a   fa
38e9 : 13 20 e7 17 ad 49 13 20   9b
38f1 : e7 17 a9 0d 20 cf 17 20   8d
38f9 : 32 30 38 08 4c 34 19 2a   72
3901 : 2a 20 41 42 42 52 55 43   67
3909 : 48 20 44 45 53 20 41 53   fd
3911 : 53 45 4d 42 4c 49 45 52   6b
3919 : 56 4f 52 47 41 4e 47 53   df
3921 : 20 2a 2a 0d 00 55 4e 54   0f
3929 : 45 52 53 54 45 20 42 59   08
3931 : 54 45 20 41 44 52 45 53   eb
3939 : 53 45 3a 20 24 00 0d 4f   d7
3941 : 42 45 52 53 54 45 20 20   55
3949 : 42 59 54 45 20 41 44 52   b7
3951 : 45 53 53 45 3a 20 24 00   f3
3959 : a0 00 20 00 02 f0 15 c9   91
3961 : 3a f0 11 c8 c9 22 d0 f2   48
3969 : 20 00 02 f0 07 c8 c9 22   4a
3971 : d0 f6 f0 e6 60 98 18 65   cb
3979 : 3d 85 3d 90 f7 e6 3e d0   2b
3981 : f3 8c 2f 13 ac 33 13 f0   7b
3989 : 2c 20 12 35 ee 33 13 a8   17
3991 : 10 07 a0 00 8c 33 13 29   4e
3999 : 7f 2c 34 13 30 0e c9 3a   ad
39a1 : d0 04 a9 00 f0 06 c9 27   92
39a9 : d0 02 a9 20 20 af 32 ac   8a
39b1 : 2f 13 c9 00 60 20 be 32   43
39b9 : a0 00 20 00 02 10 da 2c   c6
39c1 : 34 13 30 e8 8e 32 13 29   c1
39c9 : 7f a2 17 a0 44 c9 7e d0   a1
39d1 : 07 20 c5 32 a2 09 a0 46   21
39d9 : c9 4e d0 07 20 c5 32 a2   1d
39e1 : c9 a0 46 86 07 84 08 a0   53
39e9 : ff aa f0 15 c8 20 12 35   5c
39f1 : 10 fa 98 38 65 07 85 07   5e
39f9 : 90 02 e6 08 a0 ff ca d0   1c
3a01 : eb c8 8c 33 13 ae 32 13   6f
3a09 : 4c 2d 32 c9 22 d0 0a ad   de
3a11 : 34 13 49 80 8d 34 13 a9   4b
3a19 : 22 60 e6 3d d0 02 e6 3e   02
3a21 : 60 20 be 32 a0 00 20 00   12
3a29 : 02 38 e9 02 60 0a aa bd   7e
3a31 : df 32 48 bd e0 32 aa 68   0e
3a39 : 4c 10 29 21 33 38 33 45   48
3a41 : 33 64 33 77 33 89 33 ad   0a
3a49 : 33 c1 33 d3 33 fb 33 1c   bc
3a51 : 34 2d 34 45 34 5b 34 74   a9
3a59 : 34 8a 34 ab 34 ba 34 ca   d4
3a61 : 34 d1 34 21 33 21 33 21   fa
3a69 : 33 21 33 21 33 21 33 21   69
3a71 : 33 21 33 21 33 21 33 df   ef
3a79 : 34 ea 34 ff 34 42 45 46   26
3a81 : 45 48 4c 20 45 58 49 53   e4
3a89 : 54 49 45 52 54 20 4e 49   30
3a91 : 43 48 54 00 56 45 52 4d   81
3a99 : 49 53 53 45 20 27 29 27   38
3aa1 : 00 4b 45 49 4e 45 20 50   f2
3aa9 : 41 52 41 4d 45 54 45 52   be
3ab1 : 20 49 4e 20 44 45 52 20   05
3ab9 : 4b 4c 41 4d 4d 45 52 00   6d
3ac1 : 46 41 4c 53 43 48 45 52   56
3ac9 : 20 54 52 45 4e 4e 43 4f   54
3ad1 : 44 45 00 46 41 4c 53 43   cb
3ad9 : 48 45 20 50 41 52 41 4d   1c
3ae1 : 45 54 45 52 00 50 41 52   18
3ae9 : 41 4d 45 54 45 52 41 4e   35
3af1 : 4f 52 44 4e 55 4e 47 20   69
3af9 : 45 58 49 53 54 49 45 52   70
3b01 : 54 20 4e 49 43 48 54 20   2a
3b09 : 00 4f 50 45 52 41 54 49   81
3b11 : 4f 4e 20 55 4e 42 45 4b   dd
3b19 : 41 4e 4e 54 00 49 4c 4c   b4
3b21 : 45 47 41 4c 45 52 20 4f   ea
3b29 : 50 45 52 41 4e 44 00 4d   7a
3b31 : 45 48 52 20 4b 4c 41 4d   ea
3b39 : 4d 45 52 4e 20 47 45 4f   77
3b41 : 45 46 46 4e 45 54 20 41   ff
3b49 : 4c 53 20 47 45 53 43 48   bc
3b51 : 4c 4f 53 53 45 4e 00 55   f6
3b59 : 4e 5a 55 4c 41 45 53 53   e5
3b61 : 49 47 45 52 20 57 45 52   60
3b69 : 54 20 44 45 53 20 41 55   6d
3b71 : 53 44 52 55 43 4b 53 00   01
3b79 : 50 41 52 41 4d 45 54 45   01
3b81 : 52 20 3e 32 35 35 20 21   79
3b89 : 00 4c 41 42 45 4c 20 45   0a
3b91 : 58 49 53 54 49 45 52 54   9e
3b99 : 20 42 45 52 45 49 54 53   0d
3ba1 : 00 4c 41 42 45 4c 20 45   22
3ba9 : 58 49 53 54 49 45 52 54   b6
3bb1 : 20 4e 49 43 48 54 00 55   85
3bb9 : 4e 5a 55 4c 41 45 53 53   45
3bc1 : 49 47 45 52 20 4c 41 42   37
3bc9 : 45 4c 20 4e 41 4d 45 00   9a
3bd1 : 42 52 41 4e 43 48 20 46   da
3bd9 : 55 45 48 52 54 20 5a 55   88
3be1 : 20 57 45 49 54 00 55 4e   5f
3be9 : 5a 55 4c 41 45 53 53 49   f8
3bf1 : 47 45 52 20 57 45 52 54   05
3bf9 : 20 44 45 53 20 41 55 53   ff
3c01 : 44 52 55 43 4b 53 00 56   28
3c09 : 45 52 4d 49 53 53 45 20   19
3c11 : 4c 41 42 45 4c 00 49 4c   ba
3c19 : 4c 45 47 41 4c 45 52 20   7a
3c21 : 4d 4f 44 55 53 00 [illegible] 59   07
3c29 : 4e 54 41 58 00 49 4c 4c   11
3c31 : 45 47 41 4c 45 20 42 41   d5
3c39 : 4e 4b 00 42 55 53 20 46   72
3c41 : 45 48 4c 45 52 00 55 4e   7d
3c49 : 44 45 46 49 4e 49 45 52   d4
3c51 : 54 45 52 20 53 50 52 55   8c
3c59 : 4e 47 00 53 59 4d 42 4f   5d
3c61 : 4c 54 41 42 45 4c 4c 45   e2
3c69 : 20 56 4f 4c 4c 00 a9 4c   16
3c71 : 8d 00 ff b1 07 48 a9 4e   2a
3c79 : 8d 00 ff 68 60 a2 07 a0   8c
3c81 : 06 a9 00 99 02 00 88 10   f2
3c89 : fa bd 39 35 9d 00 01 ca   ca
3c91 : 10 f7 4c 00 01 a9 00 8d   28
3c99 : 00 ff 4c 09 40 a9 4e 8d   73
3ca1 : 00 ff a9 05 8d 06 d5 a9   60
3ca9 : 93 20 d2 ff a9 d1 a2 35   1f
3cb1 : 85 04 86 05 a9 0f 8d 05   ce
3cb9 : 13 a2 04 a9 0d 20 d2 ff   71
3cc1 : ca d0 f8 a2 0a 24 d7 10   c7
3cc9 : 02 a2 1e a9 20 20 d2 ff   27
3cd1 : ca d0 f8 a0 00 b1 04 f0   d5
3cd9 : 0b 20 d2 ff e6 04 d0 f3   63
3ce1 : e6 05 d0 ef e6 04 d0 02   52
3ce9 : e6 05 ce 05 13 d0 d4 a9   04
3cf1 : f0 a2 0a a0 f8 8d 08 03   eb
3cf9 : 8e 09 03 8c 04 03 8e 05   fb
3d01 : 03 a2 c0 a9 00 85 2f 86   b1
3d09 : 30 85 31 86 32 85 33 86   42
3d11 : 34 a0 0f b9 c1 35 99 f0   9e
3d19 : 0a 88 10 f7 60 a9 40 8d   da
3d21 : 00 ff 4c 39 37 a9 40 8d   38
3d29 : 00 ff 4c 6e 37 b0 c0 c0   87
3d31 : c0 c0 c0 c0 c0 c0 c0 c0   30
3d39 : c0 c0 c0 c0 c0 c0 c0 c0   38
3d41 : c0 c0 ae 0d 00 dd 20 20   5e
3d49 : 20 20 20 20 20 20 20 20   49
3d51 : 20 20 20 20 20 20 20 20   51
3d59 : 20 20 dd 0d 00 dd 20 5a   c6
3d61 : 38 30 2f 38 35 30 32 2d   7c
3d69 : 41 53 53 45 4d 42 4c 45   74
3d71 : 52 20 dd 0d 00 dd 20 20   9c
3d79 : 20 20 20 20 20 20 20 20   79
3d81 : 20 20 20 20 20 20 20 20   81
3d89 : 20 20 dd 0d 00 dd 20 20   82
3d91 : 20 20 48 41 52 41 4c 44   e5
3d99 : 20 42 45 49 4e 45 20 20   25
3da1 : 20 20 dd 0d 00 dd 20 20   9a
3da9 : 20 20 20 20 20 20 20 20   a9
3db1 : 20 20 20 20 20 20 20 20   b1
3db9 : 20 20 dd 0d 00 dd 20 20   b2
3dc1 : 20 20 20 20 20 20 20 26   cd
3dc9 : 20 20 20 20 20 20 20 20   c9
3dd1 : 20 20 dd 0d 00 dd 20 20   ca
3dd9 : 20 20 20 20 20 20 20 20   d9
3de1 : 20 20 20 20 20 20 20 20   e1
3de9 : 20 20 dd 0d 00 dd 20 20   e2
3df1 : 20 20 41 52 4e 45 20 20   8c
3df9 : 4a 41 4e 53 45 4e 20 20   69
3e01 : 20 20 dd 0d 00 dd 20 20   fa
3e09 : 20 20 20 20 20 20 20 20   09
3e11 : 20 20 20 20 20 20 20 20   11
3e19 : 20 20 dd 0d 00 dd 20 28   22
3e21 : 43 29 20 a0 44 45 5a 45   77
3e29 : 4d 42 45 52 20 31 39 38   14
3e31 : 36 20 dd 0d 00 dd 20 20   40
3e39 : 20 20 20 20 20 20 20 20   39
3e41 : 20 20 20 20 20 20 20 20   41
3e49 : 20 20 dd 0d 00 dd 20 20   42
3e51 : 20 20 20 20 42 59 20 36   69
3e59 : 34 27 45 52 20 20 20 20   80
3e61 : 20 20 dd 0d 00 dd 20 20   5a
3e69 : 20 20 20 20 20 20 20 20   69
3e71 : 20 20 20 20 20 20 20 20   71
3e79 : 20 20 dd 0d 00 ad c0 c0   b4
3e81 : c0 c0 c0 c0 c0 c0 c0 c0   80
3e89 : c0 c0 c0 c0 c0 c0 c0 c0   88
3e91 : c0 c0 bd 0d 00 a9 40 8d   2c
3e99 : 03 d5 20 80 03 c9 ac d0   72
3ea1 : 20 a5 3d 38 e9 01 85 3d   21
3ea9 : b0 02 c6 3e 20 86 03 c9   aa
3eb1 : 3a f0 0b a5 3d 38 e9 04   20
3eb9 : 85 3d b0 02 c6 3e 4c 03   df
3ec1 : 15 20 86 03 20 f1 37 4c   ef
3ec9 : ea 0a a9 40 8d 03 d5 a5   be
3ed1 : 3e 48 a5 3d 48 a9 00 a2   5b
3ed9 : 02 85 3d 86 3e 20 86 03   c3
3ee1 : aa 68 85 3d 68 85 3e 90   95
3ee9 : 49 8a c9 46 d0 1b 20 dc   d3
3ef1 : 37 a2 02 20 80 03 08 29   91
3ef9 : 7f dd aa 37 d0 2b 28 30   60
3f01 : 40 ca 10 ef 30 3b 44 4e   33
3f09 : 49 c9 4d d0 25 20 dc 37   d9
3f11 : a2 03 20 80 03 08 29 7f   61
3f19 : dd c9 37 d0 0c 28 30 2a   da
3f21 : ca 10 ef 30 25 45 47 52   33
3f29 : 45 28 a5 fc 85 3d a5 fd   60
3f31 : 85 3e 20 f1 37 4c 0d 43   ac
3f39 : a5 3d a6 3e 85 fc 86 fd   45
3f41 : 60 20 13 38 20 f1 37 4c   84
3f49 : 09 40 4c b9 38 08 a0 01   05
3f51 : 8c 03 d5 28 60 85 fc 8a   14
3f59 : 48 20 0d 43 68 aa a9 ff   e0
3f61 : a0 01 85 3d 84 3e 20 80   47
3f69 : 03 e8 e4 fc d0 f8 60 a9   63
3f71 : 04 20 f9 37 a0 00 b9 26   28
3f79 : 38 99 00 0b 88 d0 f7 4c   67
3f81 : 00 0b a9 3f 8d 00 ff a5   7d
3f89 : 2d a6 2e 85 04 86 05 a5   19
3f91 : 04 a6 05 85 61 86 62 a0   ef
3f99 : 01 b1 04 f0 42 c8 b1 04   cb
3fa1 : 85 3b c8 b1 04 85 3c a0   cb
3fa9 : 00 a9 04 85 fd b1 3d f0   74
3fb1 : 32 84 fc a4 fd d1 04 d0   19
3fb9 : 0b b1 04 f0 07 e6 fd a4   a5
3fc1 : fc c8 d0 e9 e6 04 d0 02   68
3fc9 : e6 05 a0 04 b1 04 d0 d7   09
3fd1 : a5 04 18 69 05 85 04 90   59
3fd9 : 02 e6 05 a9 00 f0 b0 8d   2a
3fe1 : 03 ff 60 a9 00 8d 00 ff   9e
3fe9 : a6 3b a5 3c 20 23 51 20   be
3ff1 : 98 55 20 b5 4b a9 3f 8d   0d
3ff9 : 00 ff a0 04 b1 04 f0 03   a6
4001 : c8 d0 f9 c8 98 18 65 04   b1
4009 : 85 04 90 02 e6 05 a9 00   32
4011 : f0 c9 4c bb 4b a9 05 20   c7
4019 : f9 37 20 ae 91 a5 2d 48   17
4021 : a5 2e 48 a9 06 8d 06 d5   b5
4029 : ad 10 12 38 e9 02 aa ac   1d
4031 : 11 12 a9 05 8d 06 d5 b0   18
4039 : 01 88 a9 00 85 b9 20 d5   3b
4041 : ff 68 85 2e 68 85 2d 98   34
4049 : 48 20 f1 37 68 a8 4c 3f   80
4051 : 91 08 78 a9 1f 8d 00 ff   98
4059 : a9 01 85 fc a9 00 85 fa   2a
4061 : a9 1c 85 fd a9 c0 85 fb   e8
4069 : a2 26 a0 00 b1 fa 91 fc   7a
4071 : c8 d0 f9 e6 fb e6 fd ca   81
4079 : d0 f2 a9 00 8d 01 1c 8d   99
4081 : 02 1c 4c 14 1c d0 f2 a9   8e
4089 : 00 8d 01 1c 8d 02 1c 4c   06
4091 : 14 1c a2 05 bd a1 40 9d   22
4099 : ea 0a ca 10 f7 4c 00 0b   35
40a1 : 8d 03 ff 4c f3 4a 44 45   66
```

Listing 1. »Double-Ass« (Schluß)

Master – Basic meisterhaft

Echte Windows, eine vielseitige RAM-Disk, Hilfen zur Behandlung von Variablen und einiges mehr bietet Ihnen »MASTER«, eine sehr komfortable Befehlserweiterung zum Abtippen.

Wo die üblichen Basic-Erweiterungen an ihre Grenzen stoßen, setzt MASTER an. Es bietet eine komfortable Ausgabe von Variablen und Feldern, macht das Generieren von Tabellen einfach und unterstützt die Fenstertechnik für menügesteuerte Programme. Neben weiteren speziellen Befehlen und Funktionen realisiert MASTER aber vor allem eine RAM-Disk, in der Programme und Daten abgelegt werden können.

Die Basic-Erweiterung MASTER knüpft an die bereits veröffentlichte Erweiterung »Butler« (64'er Sonderheft 10/86) an und ist nur zusammen mit dieser lauffähig. Sie arbeitet sowohl im 40- als auch im 80-Zeichen-Modus des C128. Eine Übersicht aller Befehle sehen Sie in Tabelle 1. MASTER profitiert von den vielen grundlegenden Möglichkeiten des Programms »Butler« und kann sich deshalb komplexeren Aufgaben zuwenden. Falls Sie Butler noch nicht besitzen, müssen Sie zusätzlich Listings 1 und 2 mit dem MSE im C64-Modus eingeben. Bitte beachten Sie dazu die Eingabehinweise auf Seite 158. Sie finden außerdem in Tabelle 2 eine kurze Übersicht über aller Befehle von »Butler«.

Das eigentliche Programm MASTER besteht aus zwei Teilen, welche Sie in Listing 3 und 4 finden. Diese müssen ebenfalls mit dem MSE im C64-Modus eingegeben werden. Der kleinere Teil liegt in einem von »Butler« bereits reservierten Bereich und der größere Teil belegt den Speicherbereich von $E000 bis $EFFF in Speicherbank 0. Zur Erinnerung sei erwähnt, daß »Butler« sowohl den dahinterliegenden Bereich von $F000 bis $FEFF als auch den von $1300 bis $1BFF beansprucht.

»Diskettenlaufwerk« im RAM

Das Bemerkenswerteste an MASTER ist sicherlich die Realisierung einer »RAM-Disk«. Dazu wird ein bestimmter Speicherbereich als »Diskette« verwendet. Auf dieser können Sie Dateien speichern, wieder laden und löschen. Dies geschieht sehr schnell, allerdings sind die Daten auf der RAM-Disk beim Ausschalten des Computers verloren. Die zur Verfügung stehende Speicherkapazität dieser RAM-Disk ist im Vergleich zu der einer Diskette natürlich relativ gering, doch soll durch sie ja kein Diskettenlaufwerk ersetzt, sondern nur sinnvoll ergänzt werden. Wer nur ein Kassettengerät besitzt, wird natürlich besonders von der RAM-Disk profitieren.

Doch MASTER bietet noch mehr als diese RAM-Disk. So ist es möglich, alle Variablen und Felder mit ihren Werten ausgeben zu lassen, um einen schnellen Überblick der verwendeten Variablen zu erhalten. Außerdem können Tabellen mit Hilfe nützlicher Befehle einfach gestaltet werden und die Programmierung von Menüs wird unterstützt. Daher lassen sich recht einfach und schnell Programme erstellen, die auch optisch einiges bieten. Zusätzlich liefert MASTER für Tüftler noch neue Befehle, mit denen selbstmodifizierende Programme erstellt und Basic-Zeilen entschlüsselt werden können. Aber dazu später mehr.

Wie oben bereits erwähnt, ist MASTER nur zusammen mit »Butler« lauffähig. Das kleine Basic-Ladeprogramm »MASTER.LADER« (Listing 5) lädt und aktiviert daher zunächst »Butler«, bevor dann mittels BOOT-Befehl die Dateien »/MASTER.1″« (Listing 3) und »/MASTER.2« (Listing 4) geladen und aufgerufen werden. Daher müssen sich alle Programmteile von »Butler« und MASTER auf



Bild 1. Ein mit »MASTER« erstelltes Menü-Fenster. Im aktiven Fenster befindet sich das Directory der RAM-Disk.

Befehle und Funktionen zur RAM-DISK (Parameter in eckigen Klammern optional)	
RAMDIR	
RAMSAVE	Name [,P Anfangsadresse TO P Endadresse+1] [,ON B Bank]
RAMLOAD	Name [,P Anfangsadresse],ON B Bank»
RAMRUN	Name
RAMSCR	Name
RAMNEW	
RAMDEV	
RAMIN	Name [,U Gerät] [,D Laufwerk]
RAMOUT	Name [,U Gerät] [,D Laufwerk]
RAMCONF	Anzahl
RAMUSED	(Nummer)
RAMFREE	(Nummer)
RAMSIZE	(Name)
RAMLOC	(Name)
RAMRES	Name,Anzahl
Variablenbehandlung	
DUMP	Liste
ARRAY	Liste
SHOW	String
Erzeugen von Menüs	
FRAME	X,Y,X_1,Y_1
VERTI	X,Y,Y_1
HORIZ	Y,X,X_1
MENUE	X,Y,X_1,Y_1
PURGE	
PURGEALL	
Sonstige Befehle	
INSERT	String
BASIC$	(String)
BASLN$	(Adresse)
MEM$	(Adresse,Anzahl)
MEMDEF	Adresse,String

Tabelle 1. Befehle und Funktionen mit den benötigten Parametern von »MASTER« im Überblick

einer Diskette befinden. Das Ladeprogramm wird am einfachsten mit »RUN "MASTER"« geladen und gestartet. Nach erfolgtem Laden ist MASTER aktiviert.

Wie schon bei »Butler«, bleibt die Erweiterung aktiv, bis die RESET-Taste betätigt oder mittels Basic-Aufruf »SYS DEC("1303")« die Abschaltung veranlaßt wird. Eine erneute Aktivierung ist mit »SYS DEC("1300")« möglich. Der Inhalt der RAM-Disk geht dabei nicht verloren.

Die Befehle von MASTER

Nach Aktivierung von MASTER stehen 40000 Bytes zur Verfügung, um Programme oder Speicherbereiche in der RAM-Disk abzulegen. Hierfür wird ein Teil des Variablenspeichers benutzt. Falls der Variablenspeicher nicht reichen sollte, kann die RAM-Disk natürlich verkleinert werden. Andererseits kann sie aber auch bis auf 58112 Bytes vergrößert werden, wenn man mit 1024 Bytes Variablenspeicher zufrieden ist.

Die RAM-Disk erlaubt es, mehrere Basic-Programme oder Programmteile im Speicher zu halten, und diese auf Wunsch blitzschnell in den Programmspeicher zu laden und automatisch zu starten. Eine andere Anwendung wäre, das gerade editierte Programm kurz in die RAM-Disk zu speichern, um einen Blick in ein anderes Programm zu werfen. Nach dem erneuten Laden kann es dann mit dem Editieren weitergehen.

Die RAM-Disk kann aber auch dazu dienen, Speicherbereiche abzulegen, um sie stets schnell zur Verfügung zu haben. Denkbar wäre es also, Textseiten, Grafikbilder, Funktionstastenbelegungen oder ähnliches auf der RAM-Disk zu sichern.

Die RAM-Disk legt die Dateien unmittelbar hintereinander in der Speicherbank 1 ab $1800 ab. Jede neue Datei wird hinter den vorhandenen im Speicher abgelegt. Beim Löschen einer Datei werden alle nachfolgenden nach vorne geschoben, um die entstandene Lücke zu schließen.

Das gilt auch für die Einträge des Inhaltsverzeichnisses der RAM-Disk, welches ebenfalls in Bank 1, jedoch gesondert ab $1200 abgelegt ist.

Jeder Eintrag im Inhaltsverzeichnis umfaßt 20 Bytes:

Byte		
0, 1	:	Dateilänge Low, High-Byte
2	:	unbenutzt
3	:	Länge des Dateinamens
4 bis 19	:	Dateiname

Insgesamt sind bis zu 12 Einträge im Inhaltsverzeichnis möglich. Diese Einschränkung ergibt sich aus programmtechnischen Gründen, um das Verzeichnis auf einer Speicherseite unterzubringen.

Alle Basic-Befehle im Zusammenhang mit der RAM-Disk beginnen mit dem Kürzel »RAM«. Ansonsten gelten für Namen und sonstige Parameter die für die Diskettenbefehle des Basic 7.0 üblichen Bezeichnungen.

Zunächst seien die Befehle vorgestellt, die dazu dienen
- Programe auf der RAM-Disk zu speichern,
- Programme von der RAM-Disk zu laden; eventuell mit automatischem Programmstart,
- einzelne oder alle Dateien auf der RAM-Disk zu löschen
- oder ein Inhaltsverzeichnis ausgeben zu lassen.

RAMDIR
Inhaltsverzeichnis auflisten (die Größe der Dateien wird in Bytes ausgegeben).

RAMSAVE "Name"
Aktuelles Basic-Programm unter dem angegebenen Namen auf der RAM-Disk speichern.

RAMSAVE "@Name"
Wie zuvor, nur wird eine eventuell existierende Datei gleichen Namens zuvor gelöscht.

Befehl	Parameter
BANKS	BankNr1,BankNr2
MEMFILL	Adresse1,Adresse2,Wert
MEMCOPY	Adresse1a,Adresse1b,Adresse2a
MEMSWAP	Adresse1a,Adresse1b,Adresse2a
MEMINV	Adresse1,Adresse2
MEMROT	Adresse1,Adresse2,Anzahl
MEMLOC	(Adresse1,Adresse2,String)
CHG	Adresse,Wert
VEC	(Adresse)
LOW	(Wert)
HIGH	(Wert)
OLD	
HIDE	[Nummer]
UNITE	
TRANS	Nummer1a,Nummer1b,Nummer2
Z80CALL	Adresse
NEWESC	Wert,Adresse
NEWCMD	Name,Adresse
NEWFN	Name,Adresse
CLRNEWS	
RESET	

Tabelle 2. Die Befehle und Funktionen der Basic-Erweiterung »Butler« aus dem Sonderheft 10/86 im Überblick. Diese funktionieren natürlich auch bei MASTER.

RAMLOAD "Name"
Datei von der RAM-Disk in den Programmspeicher laden.

RAMRUN "Name"
Datei von der RAM-Disk in den Programmspeicher laden und starten.

RAMSCR "Name"
Datei auf der RAM-Disk löschen (SCR=scratch).

RAMNEW
Alle Dateien der RAM-Disk löschen (mit Sicherheitsabfrage).

Für die meisten Anwendungen würden diese Befehle bereits ausreichen. Doch ist man erst einmal mit der RAM-Disk vertraut, wachsen auch die Ansprüche. Möglicherweise möchte man statt Programmen auch bestimmte Speicherbereiche – zum Beispiel ein Grafikbild – abspeichern und wieder laden.

Dazu dienen ebenfalls die bereits vorgestellten Befehle RAMSAVE und RAMLOAD, nur sind diese wie bei den Befehlen BSAVE und BLOAD bei Basic 7.0 um zusätzliche Parameter zu ergänzen:

RAMSAVE "Name",P Anfangadresse TO P Endadresse+1
Speicherbereich von der mittels BANK-Befehl vorgewählten Speicherbank auf der RAM-Disk ablegen.

RAMSAVE "Name",P Anfangadresse TO P Endadresse+1, ON B Bank
Wie zuvor, nur werden die Daten aus der angegebenen Speicherbank geholt, unabhängig davon, welche Bank mit dem BANK-Befehl vorgewählt wurde.

RAMLOAD "Name",P Anfangadresse
Datei aus der RAM-Disk in den Speicherbereich ab der angegebenen Adresse auf die mittels BANK-Befehl vorgewählte Speicherbank kopieren.

RAMLOAD "Name",P Anfangadresse,ON B Bank
Wie zuvor, nur werden die Daten auf die angegebene Speicherbank kopiert.

Im Gegensatz zur üblichen Speicherung auf Diskette wird bei der RAM-Disk die ursprüngliche Anfangsadresse der Daten nicht mit gespeichert. Beim Laden von Daten ist diese deshalb stets anzugeben. Fehlt sie, wird angenommen, daß es sich um ein Basic-Programm handelt und es wird in den dafür üblichen Bereich geladen. Zur Erläuterung seien zwei Beispielanwendungen angefügt:

Grafik speichern und laden:

```
BANK 0:GRAPHIC 2,1
MEMCOPY 0,8191,8192
RAMSAVE "BILD1",P8192 TO P16384
MEMCOPY 16384,24575,8192
RAMSAVE "BILD2",P8192 TO P16384
RAMLOAD "BILD1",P8192
RAMLOAD "BILD2",P8192
```

Funktionstastenbelegung für die RAM-Disk:

```
A$=CHR$(34):R$=CHR$(13)
KEY 1,"RAMLOAD"+A$+"KEYS2"+A$+",P4096,ON B0"+R$
RAMSAVE "KEYS1",P4096 TO P4351,ON B0
KEY 1,"RAMLOAD"+A$+"KEYS1"+A$+",P4096,ON B0"+R$
KEY 2,"RAMLOAD"+A$
KEY 3,"RAMDIR"+R$
KEY 4,"RAMSCR"+A$
KEY 5,"RAMSAVE"+A$
KEY 6,"RAMRUN"+A$
RAMSAVE "KEYS2",P4096 TO P4351,ON  B0
```

Über die Taste <F1> kann nun zwischen der bisherigen und der neuen Funktionstastenbelegung gewechselt werden (die jeweils nicht verwendete Funktionstastenbelegung liegt in der RAM-Disk).

Die bisherigen Befehle bezogen sich nur auf den Datentransfer innerhalb des Rechners. Natürlich ist es auch möglich, Dateien von der RAM-Disk auf Diskette oder Kassette zu speichern. Genauso kann man umgekehrt Dateien von Diskette oder Band direkt in die RAM-Disk holen. Hierfür stehen drei Befehle zur Verfügung:

RAMDEV Nummer

Verändern der voreingestellten Geräteadresse (device), auf welche sich die Befehle »RAMIN« und »RAMOUT« beziehen. Voreingestellt ist der Wert 8.
Mögliche Werte: 1,4,5,6,...,30.

RAMIN "Name"

Datei in die RAM-Disk laden. Das Überschreiben eines bereits unter dem selben Namen existierenden Files durch ein vorangestelltes »@« ist möglich. Wie bei DLOAD kann mit angehängtem Parameter »,U Nummer« ein vom voreingestellten Laufwerk verschiedenes und mit angehängtem »,D Nummer« bei einem Doppellaufwerk das gewünschte Laufwerk ausgewählt werden. Beim U-Parameter sind nur Werte von 4 bis 30 zulässig, so daß man bei Verwendung eines Kassettengeräts die Gerätenummer 1 mit dem Befehl »RAMDEV« einstellen muß (siehe oben).

RAMOUT "Name"

Datei der RAM-Disk auf dem voreingestellten Gerät speichern, das Überschreiben per Klammeraffe (»@«) ist zulässig. Für diesen Befehl gelten die selben Hinweise zur Verwendung von »U«- und »D«-Parametern, wie sie beim Befehl »RAMIN« gemacht wurden.

Nun noch ein einfaches Beispiel, an dem die Verwendung eines Kassettengeräts im Zusammenspiel mit der RAM-Disk gezeigt werden soll:

```
10 PRINT "TEST"     Beispielprogramm
RAMSAVE "TEST"      auf RAM-Disk speichern
RAMDEV 1            Kassette als Fixeinstellung
RAMOUT "TEST"       auf Kassette speichern
RAMSCR "TEST"       löschen
RAMIN "TEST"        wieder laden
RAMRUN "TEST"       und starten
RAMDEV 8            Disk als Fixeinstellung
```

Zu Beginn wurde angesprochen, daß sich die Größe der RAM-Disk den Bedürfnissen anpassen läßt. Der Befehl

RAMCONF Anzahl der Bytes

legt die Größe der RAM-Disk fest. Der sich an die RAM-Disk anschließende Variablenbereich wird neu initialisiert. Falls die neue Ausdehnung kleiner als die Anzahl der momentan verwendeten Bytes ist, wird eine Sicherheitsabfrage durchgeführt, da in diesem Fall die RAM-Disk gelöscht wird.

Zum Schluß seien noch einige spezielle Funktionen und ein zusätzlicher Befehl aufgeführt, die für geübte Programmierer sehr interessant sein dürften. Die Anweisungen »RAMUSE«, RAMFREE, RAMSIZE und RAMLOC« stellen Funktionen dar, das heißt, sie liefern einen Zahlenwert der entweder mit PRINT ausgegeben oder einer Variablen zugewiesen werden muß.

RAMUSED (Nr)

Abhängig vom Funktionsargument »Nr« liefert die Funktion folgende Daten:
Nr = 0: Größe des Programms im Programmspeicher
Nr = 1: belegter Speicher auf der RAM-Disk
Nr = 2: Anzahl belegter Einträge auf der RAM-Disk

RAMFREE (Nr)

Wiederum abhängig vom Argument »Nr«, liefert die Funktion bestimmte Daten:
Nr = 0: freie Bytes im Programmspeicher
Nr = 1: freier Speicher auf der RAM-Disk
Nr = 2: Anzahl noch möglicher Einträge auf der RAM-Disk

RAMSIZE ("Name")

Funktion, die die Größe einer Datei der RAM-Disk in Bytes liefert. Falls diese nicht existiert, ist das Ergebnis »–1«.

RAMLOC ("Name")

Funktion zur Lagebestimmung einer Datei der RAM-Disk. Liefert die aktuelle Anfangsadresse oder »–1«, wenn die angegebene Datei nicht existiert.

RAMRES "Name",Anzahl

Legt eine Datei mit der angegebenen Größe an und reserviert die gewünschte Anzahl an Bytes. Zu beachten ist, daß sich deren Lage ändert, wenn davorliegende Dateien gelöscht werden.

Variablenliste ausgeben

Eine von jedem Basic-Programmierer oft schmerzlich vermißte Möglichkeit ist die Auflistung von aktuellen Variablenwerten. Mit der Erweiterung MASTER wird dies möglich!

Je nach Bedarf können einzelne Variablen, alle Variablen eines bestimmten Typs oder einfach alle vorhandenen Variablen ausgegeben werden. Hierzu dient der Befehl

DUMP

Ohne zusätzliche Angaben werden alle Variablen ausgegeben. Möchte man über spezielle Variablen oder alle Variablen bestimmter Typen informiert werden, so sind durch Kommas getrennt die betreffenden Variablennamen bzw. die Nummern der gewünschten Variablentypen anzugeben. Die Typzuordnung ist hierfür wie folgt festgelegt:

Typnummer	:	Variablentyp
0	:	Funktionen
1	:	Fließkommavariablen
2	:	Integervariablen
3	:	Stringvariablen

Anmerkung: Funktionen lassen sich nicht einzeln mit dem Funktionsnamen sondern nur über die Typnummer ansprechen.

Am einfachsten läßt sich der Befehl »DUMP« an einigen Beispielen erläutern. Geben Sie dazu folgende Anweisungen ein:

```
CLR:XY%=-123:XYZ=1.23:XY$="123"+CHR$(13):X=1/3
```

Nun können Sie die in der nächsten Tabelle (Seite 77) angegebenen Eingaben testen.

Wie am Beispiel der Variablen »XYZ« zu ersehen ist, können Variablennamen natürlich nur an den ersten zwei Buchstaben erkannt werden, da Basic Basic 7.0 nur diese speichert.

Eingaben	Ausgaben
DUMP	XY% = -123
	XY = 1.23
	XY$ = "123"+CHR$(13)
	X = .333333333
DUMP XYZ	XY = 1.23
DUMP 3	XY$ = "123"+CHR$(13)
DUMP XY$,1,2	XY$ = "123"+CHR$(13)
	XY = 1.23
	X = .333333333
	XY% = -123

Bei der Ausgabe von String-Variablen werden einzelne Zeichen, die nicht sinnvoll oder nicht eindeutig darstellbar sind, durch die entsprechenden ASCII-Werte ausgedrückt.

Werden Zeichenketten zur Speicherung von beliebigen Daten verwendet, dann ist in der Regel stets der ASCII-Wert von Interesse. Für diesen Fall ist der Befehl

SHOW String

gedacht, der den angegebenen String als Liste der entsprechenden ASCII-Werte ausgibt. Beispiel:

```
A$=""
FOR I=0 TO 31:A$=A$+CHR$(I):NEXT
SHOW A$
```

Außer den einfachen Variablen können in Basic dimensionierte Variablen, sogenannte Felder, verwendet werden. Natürlich ermöglicht MASTER auch die Ausgabe von Feldern. Hierfür ist der Befehl

ARRAY

vorgesehen. Werden keine zusätzlichen Angaben gemacht, werden alle Felder zusammen mit der Angabe ihrer Dimensionierungen ausgegeben. Jeder zusätzlich angegebene Feldname veranlaßt die Ausgabe aller Werte des Feldes, und jede Typnummer führt zur Ausgabe der Dimensionierungen aller Felder des betreffenden Typs.

Als Typnummern sind bei Feldern nur die Werte 1 bis 3 möglich. Die Zuordnung erfolgt wie bei »DUMP«:

Typnummer	: Feldertyp
1	: Fließkomma-Felder
2	: Integer-Felder
3	: String-Felder

Die Möglichkeiten des Befehls seien wieder an einigen Beispielen aufgezeigt. Geben Sie dazu folgende Befehlsfolgen nacheinander ein:

```
CLR:DIM A(2,1):DIM A$(3):DIM A%(0,0,0,0,0,0,0,0)
FOR I=0 TO 2:FOR J=0 TO 1:A(I,J)=I-J:NEXT:NEXT
FOR K=1 TO 3:A$(K)=LEFT$("MASTER",2*K):NEXT
```

Mit dem ARRAY-Befehl kann man sich einen Überblick über die gespeicherten Felder verschaffen. Sie können die in folgender Tabelle angegebenen Eingaben testen:

Eingaben	Ausgaben
ARRAY	DIM A (2,1)
	DIM A$ (3)
	DIM A% (0,0,0,0,0,0,0,0)
ARRAY 2,3	DIM A% (0,0,0,0,0,0,0,0)
	DIM A$ (3)
ARRAY A	A(0,0) = 0
	A(1,0) = 1
	A(2,0) = 2
	A(0,1) = -1
	A(1,1) = 0
	A(2,1) = 1
ARRAY A$,A%	A$(0) = ""
	A$(1) = "MA"
	A$(2) = "MAST"
	A$(3) = "MASTER"
	A%(0,0,0,0,0,0,0,0) = 0

Die Reihenfolge der Ausgabe der Werte eines Feldes entspricht der Reihenfolge der Werte im Speicher.

Mit MASTER wird die Programmierung von Menüs und Datenfenstern einfach und elegant zugleich. Die Befehle hierfür sind schnell erklärt:

MENUE X,Y,X_1,Y_1

zeichnet auf dem aktuellen Textbildschirm ein Rechteck mit den diagonal gegenüberliegenden Eckpunkten [X,Y] (links oben) und $[X_1,Y_1]$ (rechts unten). Die darinliegende Fläche wird gelöscht und als aktuelles Fenster definiert. Dabei ist zu beachten, daß die angegebenen Eckpunkte sich auf den Rahmen und nicht auf das umschlossene Fenster beziehen. Würde man die innerhalb des Rahmens liegende Fläche mit dem »WINDOW«-Befehl des Basic 7.0 definieren, so würde dieser lauten: »WINDOW X+1,Y+1, X_1-1,Y_1-1«.

In Bild 1 sehen Sie ein mit »MENUE« erstelltes Fenster, in dem gerade das Directory der RAM-Disk sichtbar ist.

Wird das Fenster nicht mehr benötigt, läßt es sich mit

PURGE

wieder entfernen, und der ursprüngliche Bildschirminhalt wird wiederhergestellt. Dies ist möglich, weil der Inhalt in einem Teil der RAM-Disk, der gerade nicht zur Speicherung von Dateien benötigt wird, zwischengespeichert wird. Der Befehl »MENUE« setzt also voraus, daß noch genügend Platz auf der RAM-Disk frei ist, um dort vorübergehend Daten abzulegen. Die für ein Fenster erforderliche Anzahl an Bytes errechnet sich wie folgt:

$ANZAHL=2*(X_1-X+1)*(Y_1-Y+1)+54$.

Solange im Speicher genügend Platz vorhanden ist, können Fenster beliebig verschachtelt werden. Diese lassen sich entweder mit aufeinanderfolgenden PURGE-Befehlen Fenster für Fenster wieder verlassen – oder man benutzt den Befehl

PURGEALL

der dasselbe bei einmaliger Anwendung bewirkt.

Das Zusammenspiel der neuen Befehle soll folgendes Beispiel zeigen:

```
FOR I=1 TO 12:MENUE 2*I,I,10+2*I,10+I:NEXT
PURGE
PURGEALL
```

Möchte man im Textmodus Tabellen darstellen und diese schön gestalten, dann ist man in Basic 7.0 auf den »PRINT«- und den »CHAR«-Befehl angewiesen. Doch mit diesen Befehlen allein ist es umständlich und langsam, senkrechte und waagrechte Linien für Tabellen zu generieren. Im Grafikmodus dagegen wäre dies einfach, da für diesen Modus spezielle Befehle existieren.

MASTER bietet nun auch für den Textmodus Befehle an, um Linien zu ziehen oder Rechtecke zu zeichnen. Beim Befehl »MENUE« war ja bereits zu sehen, daß MASTER Rechtecke zeichnen kann, doch erledigt der Befehl »MENUE« bekanntlich noch mehr. Soll nur ein Rechteck gezeichnet werden, dann ist der Befehl

FRAME X,Y,X_1,Y_1

zu verwenden. Es wird ein Rechteck mit den Eckpunkten [X,Y] (links oben) und $[X_1,Y_1]$ (rechts unten) ausgegeben. Denkbar wäre nun, dieses Rechteck durch senkrechte und waagrechte Linien weiter zu unterteilen. Auch hierfür existieren Befehle:

VERTI X,Y,Y_1

für vertikale Linien; Zieht eine Linie vom Punkt [X,Y] zu $[X,Y_1]$. Gezeichnet wird von oben nach unten, das heißt, der Anfangspunkt muß über dem Endpunkt liegen.

HORIZ Y,X,X_1

für horizontale Linien. Zieht eine Linie vom Punkt [Y,X] zu $[Y,X_1]$. Gezeichnet wird von links nach rechts, daher muß der Anfangspunkt links vom Endpunkt liegen.

Die Tabelleneinträge für die so erzeugten Gerüste lassen sich mit dem »CHAR«-Befehl des Basic 7.0 vornehmen.

Die Rechtecke und Linien werden durch Grafikzeichen dargestellt. Diese Grafikzeichen haben allerdings im DIN-Zeichensatz andere Werte als im ASCII-Code. Master paßt sich dem an und verwendet für den beim Zeichnen aktivierten Zeichensatz die jeweils richtigen Codes.

Wie man beobachten kann, verschmelzen Grafikzeichen an Schnittpunkten von Linien zu neuen Zeichen. Beim Zeichnen von Rechtecken werden jedoch stets alle Linienabschnitte, die in das Rechteck weisen, entfernt. Daher sollte in der Regel zuerst das Rechteck gezeichnet werden, bevor die unterteilenden Linien gezogen werden. Beispiel:

```
10 SCNCLR
20 FRAME  1, 4,38,24
30 VERTI 10, 4,24
40 VERTI 30, 4,24
50 HORIZ  8, 1,38
60 HORIZ  6,10,30
70 VERTI 20, 6,24
```

Decodieren von Programmen

Bekanntlich werden Basic-Programme nicht rein als Text, sondern codiert gespeichert. Befehle und Funktionen werden als Zahlen oder Zahlenpaare abgelegt. Dies hat den Vorteil, daß Programme schneller laufen und weniger Speicherplatz benötigen. Wer jedoch Basic-Programme nicht per »LIST«-Befehl ausdrucken lassen will, weil das Listing schön formatiert werden soll, hat dadurch einen enormen Aufwand für die Decodierung.

MASTER vereinfacht diese Decodierung erheblich:

BASIC$ (String)

verwandelt einen Basic-String zurück in die leserliche Form, wie dies beim LIST-Befehl geschieht. Das Ergebnis der Entschlüsselung wird als Zeichenkette geliefert.

Ein Beispiel dazu:

```
PRINT BASIC$(CHR$(235)+CHR$(58)+CHR$(236))
```

gibt die zugehörigen Basic-Wörter der Codes »235«, »58« und »236« aus.

Es existiert aber auch eine abgewandelte Version dieser Funktion, die speziell an das Format von Basic-Zeilen angepaßt ist. Vor dem codierten Zeilentext stehen nämlich stets vier Bytes, von denen zwei die Zeilennummer darstellen. Dieser Zeilenkopf wird nach anderen Regeln decodiert und dann dem entschlüsselten Zeilentext vorangestellt. Als Parameter für die angesprochene Funktion wurde zweckmäßigerweise die Anfangsadresse der Basic-Zeile gewählt, da es zu aufwendig wäre, die codierte Basic-Zeile zunächst in eine Zeichenkette zu übertragen:

BASLN$ (Adresse)

Die Basic-Zeile wird dabei auf der mittels »BANK«-Befehl vorgewählten Speicherbank erwartet.

Ein Anwendungsbeispiel:

```
10 A=45:BANK 0
20 A=VEC(A):IF VEC(A)=0 THEN END
30 PRINT BASLN$(A):GOTO 20
```

Dieses kurze Listing decodiert sich selber und gibt das Ergebnis auf dem Bildschirm aus.

Zuvor wurde angesprochen, daß es in Basic aufwendig ist, einen Speicherbereich in eine Zeichenkette zu übernehmen. Da dies bisweilen aber nützlich sein kann, existiert auch hierfür eine Funktion:

MEM$ (Adresse,Anzahl)

Faßt ab der angegebenen Adresse die gewünschte Anzahl von Bytes zu einer Zeichenkette zusammen, die gewünschte Speicherbank muß vorgewählt werden.

Umgekehrt ist auch ein Befehl von Interesse, mit dem ein String irgendwo im Speicher plaziert werden kann:

MEMDEF Adresse,String

kopiert einen String in den Speicherbereich ab der angegebenen Adresse. Die gewünschte Bank ist vorzuwählen.

Auch zu den zwei letzten Befehlen seien wieder Beispiele angeführt:

Text aus dem Speicher holen:

```
BANK 15:PRINT MEM$(16831,9)
```

Funktionstastenbelegung neu initialisieren:

```
BANK 15:MEMDEF 4096,MEM$(52904,77)
```

Anmerkung: Bei den Anweisungen »BASIC$, BASLN$ und MEM$« handelt es sich wieder um Funktionen. Sie liefern also einen Zahlenwert der entweder mit PRINT ausgegeben, oder einer Variablen zugewiesen werden muß.

Es ist normalerweise nicht möglich, ein Basic-Programm zu schreiben, das sich während des Ablaufs selbst modifiziert - also beispielsweise neue Programmzeilen aufnimmt, alte ersetzt oder ganz löscht. Für manche Anwendungen ist dies aber recht sinnvoll oder gar notwendig. Daher bietet MASTER auch diese Möglichkeit:

INSERT String

behandelt den angegebenen String als Basic-Zeile, wertet die Zeilennummer aus, codiert den Zeilentext und fügt die entstandene Programmzeile in das laufende Programm ein. Besteht der String nur aus einer Zeilennummer, dann wird die betreffende Programmzeile gelöscht.

Einschränkungen:

Der Befehl kann nur Zeilen einfügen, löschen oder austauschen, wenn deren Zeilennummer höher ist als die der gerade ausgeführten Zeile. Ferner darf der Befehl nicht innerhalb von Unterroutinen, FOR-NEXT-Schleifen, DO-LOOP-Schleifen oder ähnlichen Strukturen liegen. Alle diese Situationen führen zu Fehlermeldungen.

Dazu ein kurzes Beispiel:

```
10 INSERT "20 LIST"
```

Mit MASTER und dem Basic 7.0 des C 128 haben sie auch ohne Assembler die Chance, umfangreiche Programme zu erstellen, welche sich durch hohe Ablaufgeschwindigkeit und ein großes Maß an Benutzerfreundlichkeit auszeichnen. Wir wünschen Ihnen viel Spaß bei der Arbeit mit dieser äußerst nützlichen Erweiterung!

(Hans-Jürgen Blechinger/Axel Pretzsch)

Eingabehinweise zu »Master«

Wenn Sie »Butler« schon aus dem Sonderheft 10 abgetippt haben, lassen Sie im Folgenden die Lade- und Speicheranweisung für »/BUTLER.2« einfach weg.

Die beiden Programme »/BUTLER.2« (Listing 2) und »/MASTER.2« (Listing 4) sind mit dem MSE im C64-Modus einzugeben und zu speichern. Anschließend muß der C128 aus- und wieder eingeschaltet werden. Laden Sie nun die beiden Programme mit

```
LOAD "/BUTLER.2",8,1
LOAD "/MASTER.2",8,1
```

im C128-Modus. Ist das geschehen, sind folgende Schritte erforderlich:

1. Mit dem Befehl MONITOR <RETURN> in den eingebauten Monitor springen.
2. Mit »T 2000 3CEB E000« die beiden Programme zusammen nach $E000 verschieben.
3. Eine leere Diskette einlegen
4. Mit
   `S "/BUTLER.2",08,F000,FCEB`
   `S "/MASERT.2",08,E000,F000`
   die beiden Programme auf Diskette speichern.
5. Die restlichen Programme (Listing 1, 3 und 5) auf die gerade bespielte Diskette kopieren.

Falls Sie die Leserservice-Diskette haben, entfallen diese Hinweise

Von nun an kann das Programm mit RUN "MASTER" geladen und gestartet werden.

```
Name : /butler.1              1300 173e
-----------------------------------------
1300 : 4c 06 13 4c 34 13 08 78   8b
1308 : 20 34 13 20 54 13 28 60   4a
1310 : 3f 7f bf ff 16 56 96 d6   1b
1318 : 2a 6a aa ea 06 0a 01 00   34
1320 : 08 48 a9 00 8d 00 ff 68   60
1328 : 28 60 08 48 a9 3f 8d 00   56
1330 : ff 68 28 60 08 78 20 20   7f
1338 : 13 20 51 42 a2 1f bd 73   f9
1340 : e0 9d 14 03 ca 10 f7 a2   a7
1348 : 09 bd 65 c0 9d 34 03 ca   be
1350 : 10 f7 28 60 20 b7 14 20   c2
1358 : c7 13 00 e2 02 20 c7 13   6c
1360 : 01 e2 02 60 00 28 43 29   00
1368 : 48 4a 42 01 06 56 d8 a9   50
1370 : 7f 8d 0d dd ac 0d dd 30   c0
1378 : 1a 20 3d f6 20 e1 ff d0   83
1380 : 12 20 56 e0 20 09 e1 20   66
1388 : 00 c0 20 06 13 20 20 13   8a
1390 : 4c 5c fa 4c 5f fa 08 48   d1
1398 : 8a 48 98 48 ba bd 05 01   25
13a0 : 85 ce bd 06 01 85 cf a0   79
13a8 : 02 fe 05 01 d0 03 fe 06   b8
13b0 : 01 b1 ce 99 e9 03 88 d0   eb
13b8 : f0 a9 01 8d ed 03 68 a8   59
13c0 : 68 aa a9 00 4c e4 03 08   f0
13c8 : 48 8a 48 98 48 ba bd 05   d6
13d0 : 01 85 ce bd 06 01 85 cf   1d
13d8 : a0 03 d0 03 99 e9 03 fe   81
13e0 : 05 01 d0 03 fe 06 01 b1   82
13e8 : ce 88 d0 f0 aa bd 10 13   4b
13f0 : 85 ce a9 01 8d ed 03 68   8c
13f8 : a8 68 aa a5 ce 4c e4 03   1c
1400 : 8d 00 ff 68 28 20 00 00   1e
1408 : 8d 01 ff 60 a2 0b bd 00   9b
1410 : 14 9d e4 03 ca 10 f7 60   5a
1418 : 8d aa 02 48 bd 10 13 8d   48
1420 : 27 14 68 60 86 79 a2 00   36
1428 : 20 a2 02 08 a6 79 28 60   b3
1430 : 8d b9 02 48 bd 10 13 8d   e7
1438 : 40 14 68 60 08 86 79 a2   88
1440 : 00 20 af 02 a6 79 28 60   14
1448 : 8d c8 02 48 bd 10 13 8d   87
1450 : 57 14 68 60 86 79 a2 00   96
1458 : 20 be 02 08 a6 79 28 60   f1
1460 : a0 ff 8d ea 03 8c eb 03   0b
1468 : a9 01 8d ed 03 60 20 7c   60
1470 : 14 8d 01 ff 60 20 7c 14   ac
1478 : 8d 02 ff 60 20 20 13 6c   3a
1480 : ea 03 20 89 14 8d 01 ff   d7
1488 : 60 8d 02 ff 6c ea 03 20   9a
1490 : 20 13 20 cf ff 8d 01 ff   ac
1498 : 60 20 20 13 20 d2 ff 8d   27
14a0 : 01 ff 60 20 20 13 20 e1   9c
14a8 : ff 8d 01 ff 60 20 20 13   5c
14b0 : 20 e4 ff 8d 01 ff 60 20   c6
14b8 : 20 13 20 7d 41 20 ee e0   ac
14c0 : 20 0c 14 20 e2 02 20 e2   74
14c8 : 02 a9 6e 8d 18 03 a9 13   53
14d0 : 8d 19 03 a9 fd 8d 3c 03   23
14d8 : a9 14 8d 3d 03 a9 42 8d   38
14e0 : 38 03 a9 15 8d 39 03 4c   ee
14e8 : b1 15 ed 14 04 00 9e ca   72
14f0 : bd 00 51 ca 14 00 3c ca   23
14f8 : 94 00 60 15 13 48 8a 29   37
1500 : 0f c9 08 68 b0 03 4c ad   b3
1508 : c6 a6 d0 e8 ec 20 0a b0   cc
1510 : 0b 9d 4a 03 a9 1b 9d 49   59
1518 : 03 e8 86 d0 4c c4 c6 aa   a6
1520 : ad ea 14 85 da ad eb 14   eb
1528 : 85 db ad ec 14 f0 11 0a   c5
1530 : 0a a8 8a 8d 01 ff 88 d1   b8
1538 : da f0 06 88 88 88 d0 f6   1b
1540 : 18 60 20 1f 15 b0 06 20   a4
1548 : 20 13 4c c1 c9 a9 02 48   c0
1550 : a9 dc 48 88 b1 da 48 88   af
1558 : b1 da 48 88 b1 da 4c df   7c
1560 : 02 46 ef a6 e4 86 eb 4c   21
1568 : b1 cb a4 e6 84 ec 20 7d   30
1570 : c7 20 8d 15 20 63 c3 4c   12
1578 : 4e ca a4 e6 84 ec 20 7d   5c
1580 : c7 20 b5 cb 20 99 15 20   a1
1588 : b5 cb 4c 4e ca e6 eb 20   d3
1590 : 74 cb b0 f9 c6 eb 4c 5c   0b
1598 : c1 a6 e5 e4 eb b0 03 4c   ab
15a0 : 70 c8 24 f8 10 05 a5 e4   26
15a8 : 4c 6f c3 20 7c c3 4c 85   c3
15b0 : cb a9 21 8d fc 02 a9 16   fd
15b8 : 8d fd 02 a2 05 bd c9 15   a8
15c0 : 9d 0c 03 ca 10 f7 4c 92   95
15c8 : 16 f8 15 db 15 46 16 0b   0d
15d0 : 28 89 89 16 16 8c 8c 16   06
15d8 : 16 75 59 c9 02 90 16 e8   07
15e0 : dd cf 15 b0 10 7d d9 15   7f
15e8 : 85 79 bc d5 15 bd d7 15   dc
15f0 : a6 79 4c 6a 51 4c 32 51   96
15f8 : f0 0d 48 ac d5 15 ad d7   83
1600 : 15 20 e2 43 68 b0 [illegible] 4c   f7
1608 : 26 43 ad cf 15 69 [illegible] c5   2b
1610 : 0d a9 8a a2 ff b0 07 e8   5c
1618 : 69 27 38 ed cf 15 4c b0   19
1620 : 43 cd cf 15 b0 1d c9 02   ff
1628 : 90 19 e9 0b 20 60 16 68   4f
1630 : 68 a5 16 48 a5 17 48 20   6e
1638 : 5a 16 68 85 17 68 85 16   5f
1640 : 4c 56 79 4c 6c 79 c9 02   5d
1648 : 90 f9 cd d0 15 b0 f4 6d   e8
1650 : cf 15 e9 31 20 60 16 20   e8
1658 : 80 03 20 d0 02 4c cd 02   3a
1660 : 85 79 0a 65 79 a8 ad d1   08
1668 : 15 85 06 ad d3 15 85 07   81
1670 : a9 3f 8d 00 ff b1 06 85   cd
1678 : 02 c8 b1 06 85 04 c8 b1   0a
1680 : 06 85 03 a9 00 8d 00 ff   ab
1688 : 60 0f 06 13 52 45 53 45   7b
1690 : d4 00 4c 3d 17 00 00 a9   e4
1698 : 00 8d 9d 16 60 00 20 83   17
16a0 : 16 20 6a 15 8d 01 ff 60   a5
16a8 : 20 83 16 20 7a 15 8d 01   9c
16b0 : ff 60 20 83 16 20 58 cb   b3
16b8 : 8d 01 ff 60 20 83 16 20   88
16c0 : ee 16 8d 01 ff 60 20 83   c7
16c8 : 16 20 64 50 8d 01 ff 60   b3
16d0 : 20 20 13 20 23 51 8d 01   be
16d8 : ff 60 20 83 16 20 80 03   ea
16e0 : 8d 01 ff 60 20 83 16 20   b0
16e8 : 80 03 8d 01 ff 60 ad 9d   62
16f0 : 16 f0 03 4c 0a ef ce 9d   5f
16f8 : 16 a5 ec 85 e9 a5 eb 85   53
1700 : e8 98 48 8a 48 a9 0d 4c   36
1708 : 6e c2 08 48 8a 48 98 48   c0
1710 : ad 00 ff 48 a0 00 ba fe   b9
1718 : 06 01 d0 03 fe 07 01 bd   db
1720 : 06 01 85 ce bd 07 01 85   05
1728 : cf b1 ce f0 05 20 99 14   81
1730 : 90 e4 68 8d 00 ff 68 a8   f1
1738 : 68 aa 68 28 60 60 ff 00   1d
```

Listing 1. Der erste Teil von »Butler«. Bitte mit dem MSE (Seite 158) im C64-Modus eingeben. Das Programm »Butler« erschien bereits im 64'er Sonderheft Ausgabe 10/86, wird aber vom Programm »MASTER« benötigt. Es muß ebenso wie Listing 2 nur dann eingegeben werden, wenn Sie »Butler« nicht besitzen.

```
Name : /butler.2              3000 3ceb
-----------------------------------------
3000 : 20 2f f1 4c e2 02 48 e0   df
3008 : 00 d0 0a a9 b1 8d 1e f0   0a
3010 : 68 8d 1f f0 60 a9 2c 8d   44
3018 : 1e f0 68 4c 18 14 b1 00   3b
3020 : 60 4c 24 14 48 e0 00 d0   5f
3028 : 0a a9 91 8d 3c f0 68 8d   25
3030 : 3d f0 60 a9 2c 8d 3c f0   35
3038 : 68 4c 30 14 91 00 60 4c   88
3040 : 3c 14 a9 02 8d 50 f0 ba   c6
3048 : bd 01 01 bc 02 01 38 e9   3b
3050 : 02 b0 05 88 90 02 a0 00   98
3058 : 8d 69 f0 8c 71 f0 60 20   c8
3060 : 7b f0 08 48 ad 78 f0 18   f1
3068 : 69 00 8d 78 f0 ad 79 f0   88
3070 : 69 00 8d 79 f0 68 28 ad   ba
3078 : 77 f0 60 08 48 8a 48 98   ac
3080 : 48 ba bd 07 01 bc 08 01   8e
3088 : 85 ce 84 cf a0 03 fe 07   bb
3090 : 01 d0 03 fe 08 01 b1 ce   87
3098 : 99 76 f0 88 d0 f0 68 a8   41
30a0 : 68 aa 68 28 60 a9 2c 20   c1
30a8 : 96 13 5e 79 60 a2 0b 2c   2e
30b0 : a2 0e 2c a2 10 2c a2 16   d2
30b8 : 2c a2 17 2c a2 19 20 96   21
30c0 : 13 3c 4d 20 96 13 ef 77   f9
30c8 : 60 20 96 13 da 77 60 20   6b
30d0 : 96 13 15 88 60 20 96 13   ce
30d8 : f7 87 60 20 96 13 12 88   0a
30e0 : 60 20 96 13 0f 88 60 20   4f
30e8 : 96 13 03 88 60 20 96 13   61
30f0 : 09 88 60 20 96 13 1e 9e   11
30f8 : 60 20 96 13 f4 87 60 20   be
3100 : 96 13 d4 84 60 20 96 13   6d
3108 : c9 84 60 20 96 13 dd 77   98
3110 : 60 20 96 13 7b 87 60 20   3e
3118 : 96 13 7e 87 60 20 96 13   50
3120 : 6e 86 60 20 96 13 90 86   3f
3128 : 60 20 96 13 e3 86 60 a9   e8
3130 : 00 a0 f0 8d 5b 13 8c 5c   a7
3138 : 13 8d 12 12 8c 13 12 4c   1b
3140 : ae f1 a9 00 a0 fe 8d ea   5f
3148 : 14 8c eb 14 a9 00 8d ec   ca
3150 : 14 a9 04 a0 ed a2 14 84   9b
3158 : 24 86 25 a2 00 85 5c a9   4e
3160 : 24 20 18 14 20 77 f1 a5   ee
3168 : 24 18 69 04 85 24 90 02   33
3170 : e6 25 c6 5c d0 ee 60 a0   6d
3178 : 03 20 24 14 20 1f 15 a2   ac
3180 : 03 b0 1b ac ec 14 c0 40   2b
3188 : b0 18 ee ec 14 c8 98 0a   9b
3190 : 0a a8 20 a5 f1 20 24 14   84
3198 : 20 a5 f1 88 91 da ca 10   53
31a0 : f1 60 4c b3 f0 48 8a 48   57
31a8 : 98 aa 68 a8 68 60 20 42   53
31b0 : f1 4c 3c f4 00 ff fe 1b   a7
31b8 : 00 ff fe 1b 03 00 00 00   0b
31c0 : 00 00 01 0f 06 13 52 45   af
31c8 : 53 45 54 00 01 00 f8 f1   ab
31d0 : 43 4c 52 4e 45 57 53 00   f4
31d8 : 01 00 9c f2 4e 45 57 43   52
31e0 : 4d 44 00 01 00 9f f2 4e   d5
31e8 : 45 57 46 4e 00 01 00 75   27
31f0 : f2 4e 45 57 45 53 43 00   42
31f8 : 20 42 f1 a9 0b 8d cf 15   71
3200 : a9 27 8d d0 15 ad d1 15   ea
3208 : ac d3 15 8d d2 15 8c d4   47
3210 : 15 8d b4 f1 8c b6 f1 ad   f9
3218 : d5 15 ac d7 15 8d d6 15   e1
3220 : 8c d8 15 8d b5 f1 8c b7   9c
3228 : f1 85 24 84 25 a9 00 a8   66
3230 : 91 24 a9 05 a0 c2 a2 f1   6d
3238 : 84 5a 86 5b a2 00 85 5c   ef
3240 : a9 5a 20 18 14 a0 00 20   a8
3248 : 24 14 99 be f1 c8 c0 04   25
3250 : 90 f5 a2 00 20 24 14 f0   d9
3258 : 07 9d 00 0c e8 c8 d0 f4   b1
3260 : 8e bd f1 98 38 65 5a 85   7f
3268 : 5a 90 02 e6 5b 20 2e f3   bf
3270 : c6 5c d0 d1 60 20 f9 f0   a3
3278 : 8e c1 f1 20 e1 f0 aa 98   e9
3280 : d0 01 ca 88 8c bf f1 8e   40
3288 : c0 f1 ae d5 03 bd 10 13   2c
3290 : 8d be f1 a9 01 a0 be a2   83
3298 : f1 4c 57 f1 a2 01 2c a2   eb
32a0 : 00 8e be f1 20 11 f1 a8   79
32a8 : f0 19 c9 11 b0 18 20 c9   99
32b0 : f2 20 e1 f0 ae d5 03 8e   0b
32b8 : bf f1 8c c0 f1 8d c1 f1   21
32c0 : 4c 2e f3 4c ad f0 4c b9   b1
32c8 : f0 a8 8c bd f1 a9 24 a2   29
32d0 : 01 20 18 14 88 20 24 14   ac
32d8 : 99 00 0c 98 d0 f6 60 a9   21
32e0 : 00 99 00 0c ad d5 15 85   17
32e8 : 24 ad d7 15 85 25 a0 00   7f
32f0 : b1 24 18 f0 19 a5 3d 48   1c
32f8 : a5 3e 48 a9 00 85 3d a9   78
3300 : 0c 85 3e 20 96 13 e6 43   87
3308 : 68 85 3e 68 85 3d 60 84   9c
3310 : 79 a5 26 d0 02 c6 27 c6   80
3318 : 26 a5 26 c5 24 a5 27 e5   2b
3320 : 25 90 0a a0 00 b1 26 a4   93
3328 : 79 91 26 b0 e4 60 ad be   8f
3330 : f1 c9 02 b0 4a ad bf f1   91
3338 : c9 11 b0 43 ac bd f1 c0   20
3340 : 11 b0 83 20 df f2 90 3a   da
3348 : c8 cc bd f1 d0 31 a5 0d   6b
3350 : e9 75 cd cf 15 a9 00 2a   54
3358 : cd be f1 d0 22 a5 0d 29   f1
3360 : 7f 06 0d 18 65 0d a8 ad   e5
3368 : d1 15 85 24 ad d3 15 85   83
3370 : 25 a2 00 bd bf f1 91 24   b8
3378 : e8 c8 e0 03 90 f5 60 4c   30
3380 : bc f0 a2 01 bd b4 f1 bc   40
3388 : b6 f1 18 7d bc f1 90 01   8c
3390 : c8 48 dd b8 f1 98 48 fd   0c
3398 : ba f1 90 03 4c b3 f0 ca   8b
33a0 : 10 e2 ae b5 f1 ac b7 f1   cb
33a8 : ad be f1 d0 13 ae d6 15   77
33b0 : 8a ac d8 15 18 6d bd f1   31
33b8 : 90 03 ee d8 15 8d d6 15   e4
33c0 : 86 24 84 25 ae b5 f1 ac   d8
```

Listing 2. Der zweite Teil von »Butler«. Bitte mit dem MSE (Seite 158) im C64-Modus eingeben.

```
33c8 : b7 f1 e8 d0 01 c8 86 26   89
33d0 : 84 27 ac bd f1 20 0f f3   0f
33d8 : 38 88 b9 00 0c 90 03 09   26
33e0 : 80 18 91 24 98 d0 f2 09   43
33e8 : 80 88 91 24 ae b4 f1 ac   47
33f0 : b6 f1 ad be f1 d0 12 ae   2d
33f8 : d2 15 8a ac d4 15 18 69   b6
3400 : 03 90 03 ee d4 15 8d d2   bc
3408 : 15 86 24 84 25 ad b4 f1   70
3410 : ac b6 f1 85 26 84 27 a0   a9
3418 : 03 20 0f f3 88 b9 bf f1   a7
3420 : 91 24 98 d0 f7 68 8d b6   6a
3428 : f1 68 8d b4 f1 68 8d b7   4f
3430 : f1 68 8d b5 f1 ae be f1   e3
3438 : fe cf 15 60 a9 00 a0 fd   88
3440 : 8d d1 15 8c d3 15 a9 00   19
3448 : a0 1a 8d d5 15 8c d7 15   53
3450 : a9 00 a0 fe 8d b8 f1 8c   81
3458 : ba f1 a9 ff a0 1b 8d b9   02
3460 : f1 8c bb f1 20 fb f1 a0   af
3468 : 73 a2 f4 a9 0b 20 38 f2   17
3470 : 4c 47 f8 00 00 f0 f4 4c   92
3478 : 4f 57 00 00 00 f2 f4 48   6f
3480 : 49 47 48 00 00 00 27 f5   08
3488 : 56 45 43 00 01 00 fe f4   48
3490 : 43 48 47 00 01 00 15 f6   1c
3498 : 42 41 4e 4b 53 00 01 00   b1
34a0 : 28 f6 4d 45 4d 46 49 4c   04
34a8 : 4c 00 01 00 50 f6 4d 45   b1
34b0 : 4d 43 4f 50 59 00 01 00   16
34b8 : 81 f6 4d 45 4d 53 57 41   00
34c0 : 50 00 01 00 f2 f6 4d 45   f7
34c8 : 4d 49 4e 56 00 01 00 2a   75
34d0 : f7 4d 45 4d 52 4f 54 00   5a
34d8 : 00 00 ab f7 4d 45 4d 4c   8f
34e0 : 4f 43 00 00 00 00 00 00   d1
34e8 : 00 00 20 c9 f0 4c cf f0   bc
34f0 : 18 b0 38 08 20 ea f4 28   ed
34f8 : 90 01 a8 4c ff f0 20 db   7c
3500 : f0 a5 16 48 a5 17 48 20   c6
3508 : e1 f0 68 85 25 68 85 24   20
3510 : a9 24 ae d5 03 20 30 14   4c
3518 : 08 78 a0 01 b9 16 00 20   31
3520 : 3c 14 88 10 f7 28 60 20   0d
3528 : ea f4 a9 16 ae d5 03 20   9f
3530 : 18 14 a0 01 20 24 14 aa   63
3538 : 88 20 24 14 a8 8a 4c 05   76
3540 : f1 a5 5e 18 65 60 85 5e   cb
3548 : 90 02 e6 5f 60 a5 5e 38   9c
3550 : e5 60 85 5e b0 02 c6 5f   87
3558 : 60 a5 5a 38 e5 5c a8 85   17
3560 : 60 a5 5b e5 5d aa 05 60   26
3568 : f0 0d b0 0c a5 5a 38 e5   66
3570 : 60 85 5a b0 02 c6 5b 18   33
3578 : 60 a5 5c 38 e5 5a a8 85   a8
3580 : 60 a5 5d e5 5b aa e8 90   97
3588 : 0c a5 5c 38 e5 60 85 5c   b5
3590 : b0 02 c6 5d 38 60 a9 5a   80
3598 : 20 06 f0 98 aa a9 5e 20   bc
35a0 : 24 f0 20 59 f5 b0 1c 20   05
35a8 : 4d f5 8a f0 10 20 1e f0   0d
35b0 : 20 3c f0 c8 d0 f7 e6 5b   63
35b8 : e6 5f e8 d0 f0 20 1e f0   0c
35c0 : 20 3c f0 60 a9 5c 20 06   50
35c8 : f0 98 aa a9 5e 20 24 f0   3d
35d0 : 20 79 f5 90 18 20 4d f5   e0
35d8 : 4c df f5 c6 5d c6 5f 20   34
35e0 : 1e f0 20 3c f0 88 c0 ff   5c
35e8 : d0 f5 ca d0 ee 60 20 db   aa
35f0 : f0 84 5a 85 5b 20 e1 f0   8a
35f8 : 84 5c 85 5d 20 e1 f0 84   95
3600 : 5e 85 5f 60 20 db f0 84   b2
3608 : 5a 85 5b 20 e1 f0 84 5c   70
3610 : 85 5d 4c ed f0 20 96 13   a5
3618 : c9 6b 20 ed f0 e0 10 b0   14
3620 : 04 8e fd 03 60 4c b0 f0   58
3628 : 20 04 f6 8a ae d5 03 85   0a
3630 : 79 a9 5a 20 24 f0 20 59   15
3638 : f5 b0 14 a5 79 e0 00 f0   c0
3640 : 0b 20 3c f0 c8 d0 fa e6   55
3648 : 5b e8 d0 f5 20 3c f0 60   73
3650 : 20 ee f5 c4 5a e5 5b b0   a1
3658 : 09 ae d5 03 ac fd 03 4c   ed
3660 : 96 f5 a5 5c 38 e5 5a aa   57
3668 : a5 5d e5 5b a8 8a 18 65   ab
3670 : 5e 85 5e 98 65 5f 85 5f   62
3678 : ae d5 03 ac fd 03 4c c4   1a
3680 : f5 20 ee f5 ae d5 03 bd   21
3688 : 10 13 ae fd 03 bc 10 13   0a
3690 : 85 24 84 25 20 59 f5 b0   f3
3698 : 1a 20 4d f5 8a f0 11 8a   5e
36a0 : 48 20 b4 f6 c8 d0 fa e6   d1
36a8 : 5b e6 5f 68 aa e8 d0 ef   70
36b0 : 20 b4 f6 60 a2 3f e4 24   f4
36b8 : d0 0f e4 25 d0 0b b1 5a   ce
36c0 : aa b1 5e 91 5a 8a 91 5e   0a
36c8 : 60 a9 5a 8d aa 02 8d b9   a9
36d0 : 02 a6 24 20 a2 02 48 a9   e1
36d8 : 5e 8d aa 02 a6 25 20 a2   41
36e0 : 02 a6 24 20 af 02 a9 5e   b1
36e8 : 8d b9 02 68 a6 25 20 af   53
36f0 : 02 60 20 db f0 84 5a 85   4d
36f8 : 5b 20 e1 f0 84 5e 85 5f   0a
3700 : ae d5 03 bd 10 13 85 24   09
3708 : 85 25 a0 00 f0 11 20 b4   c9
3710 : f6 e6 5a d0 02 e6 5b a5   3a
3718 : 5e d0 02 c6 5f c6 5e a5   29
3720 : 5a c5 5e a5 5b e5 5f 90   2d
3728 : e5 60 20 db f0 8c e4 f4   b2
3730 : 8d e5 f4 20 e1 f0 8c e8   9b
3738 : f4 8d e9 f4 20 e1 f0 aa   36
3740 : 98 18 6d e4 f4 8d e6 f4   1d
3748 : 8a 6d e5 f4 8d e7 f4 ae   ea
3750 : d5 03 bd 10 13 85 24 85   11
3758 : 25 ac e8 f4 ad e9 f4 84   b3
3760 : 5e 85 5f cc e6 f4 ed e7   90
3768 : f4 90 3f ad e6 f4 ac e7   c2
3770 : f4 85 5a 84 5b 20 0a f7   1d
3778 : ad e4 f4 ac e5 f4 85 5a   3b
3780 : 84 5b ae e7 f4 ac e6 f4   95
3788 : d0 01 ca 88 84 5e 86 5f   b1
3790 : 20 0a f7 ad e4 f4 ac e5   dd
3798 : f4 85 5a 84 5b ad e8 f4   27
37a0 : ac e9 f4 85 5e 84 5f 20   f6
37a8 : 0a f7 60 20 ea f4 84 5a   e7
37b0 : 85 5b 20 e1 f0 84 5c 85   d7
37b8 : 5d 20 a5 f0 20 c3 f0 20   d1
37c0 : 1d f1 20 c9 f2 20 de f7   b3
37c8 : 20 13 f8 a0 00 98 90 0b   e1
37d0 : a5 5a 38 ed bd f1 a8 a5   c7
37d8 : 5b e9 00 4c 05 f1 a2 01   1e
37e0 : a0 00 98 9d 00 0d ec bd   f2
37e8 : f1 b0 27 98 f0 08 bd ff   54
37f0 : 0b d9 ff 0b d0 15 e8 c8   34
37f8 : bd ff 0b d9 ff 0b d0 04   56
3800 : b9 00 0d 24 98 9d 00 0d   12
3808 : 4c e6 f7 b9 00 0d a8 4c   a0
3810 : e6 f7 60 ae d5 03 a9 5a   b1
3818 : 20 06 f0 a2 00 a0 00 ec   ab
3820 : bd f1 e8 b0 21 a5 5c c5   62
3828 : 5a a5 5d e5 5b 90 17 20   40
3830 : 1e f0 dd ff 0b f0 06 bd   09
3838 : 00 0d aa d0 f2 e6 5a d0   f5
3840 : de e6 5b d0 da 18 60 a9   c5
3848 : 06 a0 53 a2 f8 20 57 f1   99
3850 : 4c 43 fb 3f f1 fa 1d 3f   0e
3858 : 1e fb 9d 3f 65 fa 11 3f   b4
3860 : 7a fa 91 3f a7 fa 0a 3f   9d
3868 : 85 fa 84 00 00 8c 6b f8   8f
3870 : 8c 6c f8 60 20 78 f8 8a   3b
3878 : ac 6b f8 99 00 0d c8 8c   f0
3880 : 6b f8 cc 6c f8 d0 18 ee   7c
3888 : 6c f8 38 60 20 90 f8 aa   4a
3890 : ac 6b f8 cc 6c f8 f0 f2   02
3898 : 88 8c 6b f8 b9 00 0d 18   60
38a0 : 60 20 96 13 50 c1 20 e6   7a
38a8 : f8 90 0c a5 eb c5 e4 b0   82
38b0 : 0a 20 9e 16 4c a6 f8 a5   5e
38b8 : 16 a4 17 60 20 96 13 61   b8
38c0 : 15 20 e6 f8 90 f1 a5 e5   b9
38c8 : c5 eb b0 ef 20 a8 16 4c   e5
38d0 : c1 f8 20 bc 16 4c db f8   d2
38d8 : 20 b2 16 c9 3a b0 06 38   c2
38e0 : e9 30 38 e9 d0 60 20 d8   6f
38e8 : f8 b0 3e 20 97 16 20 d2   1c
38f0 : f8 b0 36 a2 00 9d 00 02   13
38f8 : e8 e0 06 b0 2c 20 d2 f8   e9
3900 : 90 f3 e9 0d 8d 2a f9 a9   0b
3908 : 00 9d 00 02 a2 ff a0 01   c6
3910 : a5 3d 48 a5 3e 48 86 3d   d5
3918 : 84 3e 20 da 16 20 96 13   02
3920 : a0 50 68 85 3e 68 85 3d   6b
3928 : 18 60 00 00 00 20 6d f8   19
3930 : a5 2d a6 2e 85 61 86 62   1d
3938 : 20 74 f8 a0 01 b1 61 f0   e9
3940 : 1e a0 03 b1 61 c5 17 90   67
3948 : 09 d0 14 88 b1 61 c5 16   39
3950 : b0 0b a0 01 b1 61 aa 88   b0
3958 : b1 61 4c 34 f9 f0 01 18   af
3960 : 60 e8 8e 2b f9 20 2d f9   87
3968 : a9 00 8d 2c f9 2a 49 01   12
3970 : 48 d0 0a ad 2a f9 d0 05   18
3978 : ce 2b f9 f0 14 ce 2c f9   d5
3980 : 20 8c f8 b0 0c 85 61 86   ba
3988 : 62 ee 2c f9 ce 2b f9 d0   7b
3990 : ef ae 2c f9 68 60 a9 00   51
3998 : 8d 2c f9 8e 2b f9 20 c6   1c
39a0 : 16 08 ae 2b f9 f0 29 90   b8
39a8 : 06 ad 2a f9 f0 01 e8 a9   5d
39b0 : 00 8d 2c f9 ca f0 19 8e   77
39b8 : 2b f9 a0 01 b1 61 aa f0   db
39c0 : 0f 88 b1 61 85 61 86 62   ee
39c8 : ee 2c f9 ce 2b f9 d0 ea   c0
39d0 : ae 2c f9 28 a0 01 b1 61   b3
39d8 : 60 8d 0e fa 4c f1 f9 20   5e
39e0 : 15 fa 20 9e 16 a0 00 b1   18
39e8 : 61 aa c8 b1 61 86 61 85   e1
39f0 : 62 a0 01 b1 61 f0 54 20   48
39f8 : a3 14 f0 4f ad 0e fa f0   e4
3a00 : 06 ce 0e fa 4c df f9 a5   47
3a08 : eb c5 e4 90 d2 60 00 a0   92
3a10 : 01 b1 61 f0 36 a0 02 b1   34
3a18 : 61 aa c8 b1 61 20 d0 16   bd
3a20 : 20 96 13 76 ca a5 ec c5   38
3a28 : e6 f0 11 a5 eb 48 20 96   2e
3a30 : 13 8d 15 68 c5 eb 85 eb   06
3a38 : b0 0f e6 eb 20 96 13 5c   63
3a40 : c1 20 96 13 55 ca 20 a8   97
3a48 : 16 a9 01 60 85 16 84 17   c8
3a50 : 4c c6 16 48 a5 e4 38 e5   bc
3a58 : e5 aa 68 60 48 a5 e4 38   6e
3a60 : e5 e5 4a aa 68 60 20 bc   a3
3a68 : f8 b0 30 20 53 fa 20 96   83
3a70 : f9 f0 4f a2 01 20 96 f9   69
3a78 : 4c b6 fa 20 a1 f8 b0 42   0b
3a80 : 20 53 fa 4c e2 fa ae 08   63
3a88 : 12 e8 f0 0e ad 09 12 ac   d1
3a90 : 0a 12 20 4c fa 90 03 20   b5
3a98 : d7 fa 60 38 6e 2a f9 a9   7f
3aa0 : 00 a8 20 4c fa 4c b6 fa   69
3aa8 : 20 a1 f8 b0 ee 20 c6 16   24
3ab0 : a0 01 b1 61 f0 0c 20 96   86
3ab8 : 13 42 c1 46 f0 a9 00 4c   1a
3ac0 : d9 f9 38 6e 2a f9 20 53   0b
3ac8 : fa a9 ff a8 85 16 84 17   f5
3ad0 : 4c e2 fa 85 16 84 17 20   1f
3ad8 : 5c fa 20 96 f9 f0 e3 20   83
3ae0 : 5c fa 20 61 f9 8a 48 20   43
3ae8 : 96 13 42 c1 46 f0 68 4c   f7
3af0 : d9 f9 46 f0 20 bc f8 b0   a3
3af8 : a2 20 9e 16 a2 01 20 96   f5
3b00 : f9 d0 0b 8d 2a f9 a2 01   d5
3b08 : 20 61 f9 4c 13 fb ad 2a   fd
3b10 : f9 d0 03 20 a8 16 20 0f   10
3b18 : fa f0 03 20 9e 16 60 46   f8
3b20 : f0 20 a1 f8 b0 9c 20 a8   69
3b28 : 16 a2 01 20 61 f9 d0 09   0f
3b30 : ad 2a f9 f0 04 90 05 b0   c9
3b38 : 09 20 9e 16 20 0f fa 20   62
3b40 : a8 16 60 a0 52 a2 fb a9   9d
3b48 : 05 20 38 f2 20 83 fc 4c   74
3b50 : db fc 01 00 83 fb 4f 4c   d8
3b58 : 44 00 01 00 93 fb 48 49   a9
3b60 : 44 45 00 01 00 ac fb 55   67
3b68 : 4e 49 54 45 00 01 00 c4   aa
3b70 : fb 54 52 41 4e 53 00 01   d4
3b78 : 00 63 fc 5a 38 30 43 41   49
3b80 : 4c 4c 00 a9 01 a8 91 2d   1e
3b88 : 20 96 13 4f 4f 20 96 13   19
3b90 : 82 4f 60 d0 08 a9 ff 85   c5
3b98 : 16 85 17 d0 03 20 db f0   d3
3ba0 : 20 c6 16 a5 61 a4 62 85   2d
3ba8 : 2d 84 2e 60 a9 1c a0 01   af
3bb0 : a6 76 f0 03 18 69 24 84   94
3bb8 : 2d 85 2e 4c 88 fb 20 96   d3
3bc0 : 13 f0 84 60 20 be fb 20   a1
3bc8 : 47 fc 8c e4 f4 8d e5 f4   8a
3bd0 : 20 a5 f0 20 47 fc 20 52   84
3bd8 : fc a4 61 a5 62 8c e6 f4   43
3be0 : 8d e7 f4 cc e4 f4 ed e5   b1
3be8 : f4 90 59 20 a5 f0 20 47   70
3bf0 : fc 8c e8 f4 8d e9 f4 cc   a1
3bf8 : e6 f4 ed e7 f4 b0 32 ac   c8
3c00 : e4 f4 ad e5 f4 cc e8 f4   ca
3c08 : ed e9 f4 90 37 ae e8 f4   af
3c10 : ac e9 f4 ad e6 f4 8d e8   c2
3c18 : f4 ad e7 f4 8d e9 f4 ad   d3
3c20 : e4 f4 8d e6 f4 ad e5 f4   fd
3c28 : 8d e7 f4 8e e4 f4 8c e5   ac
3c30 : f4 ad e8 f4 d0 03 ce e9   08
3c38 : f4 ce e8 f4 a2 00 20 52   bb
3c40 : f7 4c 88 fb 4c b0 f0 20   4d
3c48 : db f0 20 c6 16 a4 61 a5   d4
3c50 : 62 60 90 0e a0 01 b1 61   64
3c58 : aa f0 07 88 b1 61 85 61   4c
3c60 : 86 62 60 20 db f0 8c 9c   e4
3c68 : fc 8d 9d fc 08 78 ad 00   2d
3c70 : ff 48 a9 3e 8d 00 ff a9   f2
3c78 : b0 8d 05 d5 ea 68 8d 00   13
3c80 : ff 28 60 a9 c3 8d ee ff   45
3c88 : a9 93 8d ef ff a9 fc 8d   b8
3c90 : f0 ff 60 31 d0 ff 3e 3f   43
3c98 : 32 00 ff cd 00 00 c3 e0   55
3ca0 : ff a2 01 8e ca fc a9 30   9e
3ca8 : 8d cf fc a0 ca a2 fc a9   79
3cb0 : 01 20 38 f2 ee cf fc ae   ec
3cb8 : ca fc bd cf 15 dd c8 fc   c7
3cc0 : 90 e9 ce ca fc f0 e4 60   fd
3cc8 : 1b 57 00 0f e2 02 40 40   31
3cd0 : 40 00 28 43 29 48 4a 42   05
3cd8 : 01 06 56 20 c7 13 00 e2   51
3ce0 : 02 20 c7 13 01 e2 02 20   b6
3ce8 : a1 fc 60 00 86 00 86 79   95
```

Listing 2. »/BUTLER.2«
Der zweite Programmteil (Schluß)

```
Name : /master.1              173d 1800
---------------------------------------
173d : 4c ef 17 20 83 16 20 5c   6d
1745 : c1 20 58 cb 8d 01 ff 60   47
174d : 48 20 83 16 20 5c c1 68   06
1755 : 20 32 cc 8d 01 ff 60 00   05
175d : 00 00 00 00 ad 02 03 8d   70
1765 : 5c 17 ad 03 03 8d 5d 17   59
176d : a9 8a 8d 02 03 a9 17 8d   f4
1775 : 03 03 a5 3d 8d 00 12 a5   77
177d : 3e 8d 01 12 a9 ff 85 3d   30
1785 : a9 0b 85 3e 60 ad 5c 17   f0
178d : 8d 02 03 ad 5d 17 8d 03   5d
1795 : 03 a5 7f d0 03 6c 02 03   06
179d : ad 00 12 85 3d ad 01 12   e9
17a5 : 85 3e 4c f6 4a 85 ef 98   fd
17ad : 48 8a 48 ae 60 17 a5 ef   57
17b5 : 9d 00 0d ee 60 17 f0 08   06
17bd : 68 aa 68 a8 a5 ef 18 60   a4
17c5 : 20 e2 17 4c ed a5 a9 00   58
17cd : 8d 60 17 ad 26 03 8d 5e   73
17d5 : 17 ad 27 03 8d 5f 17 a2   63
17dd : aa a0 17 d0 06 ae 5e 17   35
17e5 : ac 5f 17 8e 26 03 8c 27   d3
17ed : 03 60 20 83 16 24 d7 10   9b
17f5 : 06 20 b3 77 4c ff 17 20   48
17fd : c4 77 60 ff 00 ff 00 ff   95
```

Listing 3. Der erste Teil des Programms »MASTER«. Bitte mit dem MSE (Seite 158) im C64-Modus eingeben.

```
Name : /master.2              2000 3000
---------------------------------------
2000 : a9 86 8d e8 16 a9 a9 8d   dd
2008 : ea f3 a9 1a a0 e0 8d df   a0
2010 : fc 8c e0 fc 20 cd e4 4c   c7
2018 : 00 f0 a9 00 a0 e0 8d 12   66
2020 : 12 8c 13 12 a0 30 a2 e0   57
2028 : a9 04 20 38 f2 4c d0 e4   81
2030 : 01 00 a3 e0 44 55 4d 50   fb
2038 : 00 01 00 d8 e0 41 52 52   da
2040 : 41 59 00 01 0f e2 02 40   df
2048 : 23 40 00 01 00 85 e0 53   02
2050 : 48 4f 57 00 a9 61 2c a9   bf
2058 : 5a 2c a9 16 a2 01 4c 18   89
2060 : 14 20 11 f1 a6 24 a4 25   6f
2068 : 86 5a 84 5b 85 5c a9 5a   3e
2070 : a2 01 20 18 14 a0 00 a5   30
2078 : 5c 60 84 59 a8 a9 00 20   69
2080 : ad e4 a4 59 60 20 61 e0   42
2088 : f0 13 20 24 14 c8 20 7a   8b
2090 : e0 c4 5c b0 05 20 96 e4   75
2098 : 90 f0 20 88 e4 20 96 13   89
20a0 : 98 55 60 f0 16 20 09 e3   67
20a8 : 90 06 20 07 e1 4c c0 e0   6a
20b0 : 20 f9 f0 e0 04 b0 1e bd   df
20b8 : 6d e3 2c a9 80 20 35 e1   f9
20c0 : 90 10 20 e4 16 c9 2c d0   ff
20c8 : 06 20 da 16 4c a5 e0 20   0d
20d0 : 9a e0 4c 88 e4 4c b0 f0   54
20d8 : f0 19 20 09 e3 90 06 20   99
20e0 : 16 e2 4c f8 e0 20 f9 f0   72
20e8 : 8a f0 ea e0 04 b0 e6 bd   9e
20f0 : 6d e3 2c a9 80 20 79 e1   42
20f8 : 90 d8 20 e4 16 c9 2c d0   9b
2100 : ce 20 da 16 4c da e0 20   b7
2108 : a3 14 f0 27 20 a6 e2 20   d9
2110 : 23 e3 20 96 13 7b 79 20   33
2118 : f6 e2 20 99 e3 20 0a 17   50
2120 : 20 3d 20 00 24 0f 10 06   ee
2128 : 20 10 e4 4c 31 e1 20 b0   17
2130 : e4 38 60 18 60 85 4b 20   eb
2138 : 5a e0 a6 2f a5 30 86 16   b4
2140 : 85 17 e4 31 e5 32 b0 30   c3
2148 : 20 a3 14 f0 e6 20 79 e3   7a
2150 : 08 a5 16 18 69 02 85 16   9c
2158 : 90 02 e6 17 28 d0 0d 20   03
2160 : 99 e3 20 0a 17 20 3d 20   dc
2168 : 00 20 b8 e3 a5 16 18 69   61
2170 : 05 aa a5 17 69 00 90 c6   7d
2178 : 60 85 4b 20 5a e0 a6 31   1b
2180 : a5 32 86 16 85 17 e4 33   ae
2188 : e5 34 b0 47 20 a3 14 f0   ee
2190 : 41 20 79 e3 d0 37 20 9a   39
2198 : e0 20 0a 17 44 49 4d 20   f2
21a0 : 00 20 9c e3 20 e4 e1 a9   58
21a8 : 28 2c a9 2c 20 99 14 86   02
21b0 : 5d bc 40 0c bd 00 0c 38   79
21b8 : e9 01 aa 98 e9 00 20 a7   4e
21c0 : e4 a6 5d e8 e4 5e 90 e2   b5
21c8 : a9 29 20 99 14 20 d4 e1   9a
21d0 : 90 b0 18 60 a0 02 18 20   85
21d8 : 24 14 65 16 aa c8 20 24   dc
21e0 : 14 65 17 60 a0 04 20 24   6c
21e8 : 14 f0 22 c9 41 b0 1e 85   53
21f0 : 5e aa ca c8 20 24 14 9d   1e
21f8 : 40 0c c8 20 24 14 9d 00   ce
2200 : 0c a9 00 9d c0 0c 9d 80   78
2208 : 0c 8a d0 e6 60 a2 1a 4c   86
2210 : be f0 38 60 18 60 20 a6   b3
2218 : e2 20 50 e3 90 f4 20 e4   96
2220 : e1 c8 98 d0 05 a6 4c bd   d8
2228 : 75 e3 18 65 16 85 16 90   49
2230 : 02 e6 17 20 a3 14 f0 dc   c8
2238 : 20 9a e0 a5 47 29 7f 20   8e
2240 : 99 14 a5 48 29 7f f0 03   ae
2248 : 20 99 14 a6 4c f0 08 bd   f7
2250 : 71 e3 20 99 14 a2 00 a9   98
2258 : 28 2c a9 2c 20 99 14 86   b2
2260 : 5d bd c0 0c a8 bd 80 0c   e0
2268 : aa 98 20 a7 e4 a6 5d e8   26
2270 : e4 5e 90 e6 20 0a 17 29   85
2278 : 20 3d 20 00 20 b8 e3 a2   dc
2280 : 00 fe 80 0c d0 03 fe c0   44
2288 : 0c bd 80 0c dd 00 0c bd   9e
2290 : c0 0c fd 40 0c 90 8e a9   b1
2298 : 00 9d 80 0c 9d c0 0c e8   ea
22a0 : e4 5e 90 dd 38 60 a2 00   a4
22a8 : 86 0f 86 10 20 e4 16 85   e6
22b0 : 47 20 09 e3 b0 03 4c ad   76
22b8 : f0 20 da 16 90 05 20 09   f5
22c0 : e3 90 0b aa 20 da 16 90   56
22c8 : fb 20 09 e3 b0 f6 c9 24   c4
22d0 : d0 06 a9 ff 85 0f d0 0c   3a
22d8 : c9 25 d0 0f a9 80 85 10   1f
22e0 : 05 47 85 47 8a 09 80 aa   1b
22e8 : 20 da 16 86 48 20 e4 16   11
22f0 : f0 04 c9 2c d0 c0 a9 00   94
22f8 : 85 4c a5 47 0a 26 4c 4a   8d
2300 : aa a5 48 0a 26 4c 4a a8   0f
2308 : 60 c9 41 90 05 e9 5b 38   2d
2310 : e9 a5 60 a0 00 20 24 14   b2
2318 : c8 c5 47 d0 05 20 24 14   b9
2320 : c5 48 60 20 54 e0 a6 2f   6b
2328 : a5 30 86 61 85 62 e4 31   14
2330 : e5 32 b0 11 20 13 e3 f0   89
2338 : 11 a5 61 18 69 07 aa a5   3c
2340 : 62 69 00 90 e5 a9 d2 a0   a1
2348 : 03 60 20 96 13 57 7c 60   f5
2350 : 20 5a e0 a6 31 a5 32 86   c0
2358 : 16 85 17 e4 33 e5 34 b0   28
2360 : 0a 20 13 e3 f0 06 20 d4   25
2368 : e1 90 ec 18 60 02 00 03   ec
2370 : 01 20 24 5e 25 05 03 00   dd
2378 : 02 a0 00 84 4c 20 24 14   d9
2380 : 0a 26 4c 4a aa c8 20 24   b3
2388 : 14 0a 26 4c 4a a8 a5 4b   cc
2390 : 30 05 29 03 c5 4c 60 0a   42
2398 : 60 20 9a e0 8a 20 99 14   03
23a0 : 98 08 f0 03 20 95 [illegible] a6   45
23a8 : 4c bd 71 e3 20 99 14 28   1b
23b0 : d0 03 20 9c e4 a0 00 60   b2
23b8 : 20 5a e0 a6 4c d0 0c a5   d9
23c0 : 16 a4 17 20 96 13 85 7a   ff
23c8 : 4c 0c e4 a0 00 ca d0 1b   37
23d0 : 20 24 14 48 c8 20 24 14   57
23d8 : aa c8 20 24 14 a8 68 20   db
23e0 : 68 e0 20 18 e4 20 5a e0   3e
23e8 : 4c 0f e4 ca d0 11 c8 20   47
23f0 : 24 14 aa 88 20 24 14 a8   9f
23f8 : 8a 20 05 f1 4c 0c e4 20   0b
2400 : 24 14 aa c8 20 24 14 a8   b7
2408 : 8a 20 a1 e4 20 b0 e4 60   83
2410 : 20 96 13 81 87 20 64 e0   3d
2418 : 84 5d c4 5c b0 51 20 24   66
2420 : 14 c9 ff f0 12 48 29 7f   3e
2428 : c9 60 68 b0 16 a2 06 dd   9c
2430 : 81 e4 f0 0f ca 10 f8 24   9b
2438 : 5d 30 05 48 20 99 e4 68   2b
2440 : 38 b0 24 24 5d 10 06 20   0d
2448 : 0a 17 22 2b 00 20 0a 17   23
2450 : 43 48 52 24 28 00 20 7a   c8
2458 : e0 c8 c4 5c 88 a9 29 b0   35
2460 : 05 20 99 14 a9 2b 18 66   7f
2468 : 5d 20 99 14 c8 d0 ab a4   c9
2470 : 5c d0 06 20 99 e4 20 99   2e
2478 : e4 24 5d 10 03 20 99 e4   29
2480 : 60 00 0d 14 1b 22 8d de   5d
2488 : a5 9a c9 03 d0 f2 a9 1b   cf
2490 : 20 99 14 a9 51 2c a9 2c   2d
2498 : 2c a9 22 2c a9 20 4c 99   a7
24a0 : 14 20 96 13 3c 79 60 20   1e
24a8 : 96 13 32 8e 60 20 05 f1   25
24b0 : 20 96 13 42 8e 85 24 84   d7
24b8 : 25 a0 00 b1 24 c9 20 d0   16
24c0 : 01 c8 b1 24 f0 06 20 99   09
24c8 : 14 c8 d0 f6 60 4c e4 e4   19
24d0 : a0 ee a2 e4 a9 0f 20 38   30
24d8 : f2 a9 08 20 f3 e6 20 0f   ba
24e0 : e6 4c 53 eb a2 40 a0 b4   57
24e8 : 20 e7 e5 4c 50 eb 01 00   67
24f0 : 24 e6 52 41 4d 4e 45 57   4f
24f8 : 00 01 00 3d e9 52 41 4d   f1
2500 : 44 49 52 00 01 00 e2 e6   e7
2508 : 52 41 4d 44 45 56 00 01   e0
2510 : 00 c9 e7 52 41 4d 53 41   87
2518 : 56 45 00 01 00 4d e8 52   e4
2520 : 41 4d 4c 4f 41 44 00 01   3d
2528 : 00 5c e7 52 41 4d 4f 55   01
2530 : 54 00 01 00 fc e6 52 41   97
2538 : 4d 49 4e 00 01 00 36 e8   78
2540 : 52 41 4d 52 55 4e 00 01   9a
2548 : 00 79 e8 52 41 4d 53 43   dc
2550 : 52 00 01 00 22 e8 52 41   18
2558 : 4d 52 45 53 00 01 00 ac   ec
2560 : e5 52 41 4d 43 4f 4e 46   dd
2568 : 00 00 00 41 e6 52 41 4d   31
2570 : 53 49 5a 45 00 00 00 61   6a
2578 : e6 52 41 4d 4c 4f 43 00   ce
2580 : 00 00 75 e6 52 41 4d 55   ca
2588 : 53 45 44 00 00 00 aa e6   07
2590 : 52 41 4d 46 52 45 45 00   03
2598 : 00 18 00 18 40 b4 40 b4   bc
25a0 : 00 00 00 00 12 00 08 00   e2
25a8 : 00 00 00 00 20 db f0 aa   a2
25b0 : 18 98 6d 98 e5 a8 8a 6d   2b
25b8 : 99 e5 aa b0 64 84 5a 86   e5
25c0 : 5b 18 98 69 ff a8 8a 69   bd
25c8 : 03 b0 56 e4 39 e5 3a b0   63
25d0 : 50 a6 5a ec 9a e5 a4 5b   ca
25d8 : 98 ed 9b e5 b0 0c 20 3b   6d
25e0 : e6 d0 57 a6 5a a4 5b 20   51
25e8 : 29 e6 8e 9e e5 8c 9f e5   09
25f0 : 20 15 e6 4c fa e5 a6 2f   b6
25f8 : a4 30 86 2f 84 30 86 31   82
2600 : 84 32 86 33 84 34 a6 39   9c
2608 : a4 3a 86 35 84 36 60 ae   ea
2610 : 9e e5 ac 9f e5 8e 9c e5   d1
2618 : 8c 9d e5 a9 00 8d a0 e5   dc
2620 : 60 4c b0 f0 20 3b e6 d0   0a
2628 : 11 a9 00 8d a5 e5 ad 98   31
2630 : e5 8d 9a e5 ad 99 e5 8d   99
2638 : 9b e5 60 20 96 13 e1 a7   bb
2640 : 60 20 1d f1 20 c1 e8 20   2a
2648 : 9a ea d0 0c 20 02 ea ac   24
2650 : b3 12 ad b4 12 4c 05 f1   8a
2658 : a9 ff a8 20 96 13 3c 79   15
2660 : 60 20 1d f1 20 c1 e8 20   4a
2668 : 9a ea d0 ec ac a1 e5 ad   14
2670 : a2 e5 4c 05 f1 20 d5 f0   12
2678 : 8a d0 0f 38 ad 10 12 e5   a5
2680 : 2d a8 ad 11 12 e5 2e 4c   30
2688 : 05 f1 ca d0 11 38 ad 9a   11
2690 : e5 ed 98 e5 a8 ad 9b e5   81
2698 : ed 99 e5 4c 05 f1 ca d0   01
26a0 : 06 ac a5 e5 4c ff f0 4c   43
26a8 : b0 f0 20 d5 f0 8a d0 11   5c
26b0 : 38 ad 12 12 ed 10 12 a8   7f
26b8 : ad 13 12 ed 11 12 4c 05   0e
26c0 : f1 ca d0 11 38 ad 9c e5   9c
26c8 : ed 9a e5 a8 ad 9d e5 ed   cc
26d0 : 9b e5 4c 05 f1 ca d0 cf   6a
26d8 : a9 0c 38 ed a5 e5 a8 4c   18
26e0 : ff f0 20 f9 f0 8a f0 0f   e4
26e8 : ca f0 08 c9 04 90 08 c9   de
26f0 : 1f b0 04 8d a6 e5 60 a2   7a
26f8 : 09 4c be f0 a9 7f 20 7f   0b
2700 : e7 20 e8 e8 a9 00 ae 9a   d9
2708 : e5 ac 9b e5 20 96 13 d5   96
2710 : ff 08 20 96 13 43 92 28   d4
2718 : b0 32 20 96 13 51 92 29   15
2720 : bf d0 2e 8e aa e5 8c ab   08
2728 : e5 8a 38 ed 9a e5 8d b3   95
2730 : 12 98 ed 9b e5 8d b4 12   3f
2738 : ad 9c e5 cd aa e5 ad 9d   32
2740 : e5 ed ab e5 90 10 20 2a   22
2748 : e9 4c c7 ea 20 96 13 d0   4b
2750 : 90 20 96 13 67 91 20 f6   6a
2758 : e5 4c a2 e8 a9 ff 20 7f   43
2760 : e7 20 91 e8 ad a1 e5 85   63
2768 : 5a 18 6d b3 12 aa ad a2   13
2770 : e5 85 5b 6d b4 12 a8 a9   6e
2778 : 5a 20 96 13 1d 91 60 48   5b
2780 : a9 66 a2 ff 20 aa e8 a2   45
2788 : 00 8e 1d 01 a9 01 20 96   87
2790 : 13 68 ff 68 25 80 85 80   52
2798 : 29 08 d0 23 ae a6 e5 8e   33
27a0 : 1c 01 e0 04 b0 19 a5 80   61
27a8 : 29 90 d0 73 a8 20 96 13   c8
```

Listing 4. Der zweite Teil von »MASTER«. Bitte mit dem MSE (Seite 158) eingeben.

```
27b0 : ba ff ad b6 12 a2 b7 a0  02
27b8 : 12 20 96 13 bd ff 60 a0  81
27c0 : 05 a9 04 20 96 13 67 a6  8c
27c8 : 60 a9 66 a2 f8 20 aa e8  f8
27d0 : a5 81 d0 15 20 e8 e8 a4  43
27d8 : 2d a6 2e 84 5a 86 5b ac  15
27e0 : 10 12 ae 11 12 a9 00 f0  17
27e8 : 1c 29 06 c9 06 d0 30 20  3c
27f0 : e8 e8 ac 17 01 ae 18 01  42
27f8 : 84 5a 86 5b ac 19 01 ae  ab
2800 : 1a 01 ad 1f 01 84 5c 86  9d
2808 : 5d 48 38 98 e5 5a a8 8a  93
2810 : e5 5b 90 08 20 04 e9 68  63
2818 : aa 4c b8 ea 4c b0 f0 4c  1a
2820 : ad f0 20 be e8 20 e1 f0  1e
2828 : 20 e8 e8 a4 16 a5 17 20  b6
2830 : 04 e9 20 c7 ea 60 20 8e  79
2838 : e8 20 ce ea 8e 10 12 8c  0c
2840 : 11 12 68 68 a9 5a 48 a9  63
2848 : a5 48 4c dd 02 a9 e6 a2  2e
2850 : fc 20 aa e8 20 91 e8 a5  a1
2858 : 81 f0 10 29 02 f0 c0 ac  7f
2860 : 17 01 ad 18 01 ae 1f 01  6a
2868 : 4c d4 ea 20 ce ea 68 68  94
2870 : a9 91 48 a9 6b 48 4c dd  0f
2878 : 02 20 be e8 20 9a ea d0  7b
2880 : 1b 20 02 ea 8a 48 20 f8  e7
2888 : ea 68 aa 4c 86 ea 20 be  98
2890 : e8 20 9a ea d0 06 4c 02  ff
2898 : ea a2 01 2c a2 04 2c a2  d9
28a0 : 08 2c a2 10 2c a2 1a 4c  42
28a8 : be f0 20 96 13 c3 a3 20  d7
28b0 : 96 13 50 a7 8d a7 e5 ae  e4
28b8 : 11 01 8e b6 12 60 20 11  8b
28c0 : f1 a2 00 86 80 20 96 13  5d
28c8 : bf a5 8d b6 12 a5 80 8d  ff
28d0 : a7 e5 a9 24 a2 01 20 18  3c
28d8 : 14 a0 00 20 24 14 99 b7  f9
28e0 : 12 c8 cc b6 12 90 f4 60  9a
28e8 : 20 9a ea d0 0c 2c a7 e5  d7
28f0 : 10 b3 20 ea e9 20 7c e8  a3
28f8 : 20 f6 e9 ae a5 e5 e8 e0  d2
2900 : 0d b0 96 60 8c b3 12 8d  e1
2908 : b4 12 a8 98 18 6d 9a e5  26
2910 : 8d aa e5 98 6d 9b e5 8d  e5
2918 : ab e5 b0 1e ad 9c e5 cd  99
2920 : aa e5 ad 9d e5 ed ab e5  24
2928 : 90 10 ae a5 e5 20 18 ea  b6
2930 : ee a5 e5 ad aa e5 ac ab  04
2938 : e5 60 4c a2 e8 20 0a 17  9b
2940 : 12 3e 3e 3e 20 20 20 20  8d
2948 : 20 52 41 4d 20 44 49 53  7b
2950 : 4b 20 20 20 20 20 3c 3c  24
2958 : 3c 00 20 9a e0 a2 00 ec  ed
2960 : a5 e5 b0 09 20 02 ea 20  43
2968 : c0 e9 e8 d0 f2 38 ad 9c  52
2970 : e5 ed 9a e5 aa ad 9d e5  09
2978 : ed 9b e5 a8 8a 20 93 e9  8d
2980 : 20 0a 17 20 42 59 54 45  3a
2988 : 53 20 46 52 45 45 2e 00  fe
2990 : 4c 88 e4 aa e0 10 98 e9  73
2998 : 27 b0 1e 20 9c e4 e0 e8  e9
29a0 : 98 e9 03 b0 14 20 9c e4  82
29a8 : e0 64 98 e9 00 b0 0a 20  0c
29b0 : 9c e4 e0 0a b0 03 20 9c  15
29b8 : e4 98 20 96 13 32 8e 60  81
29c0 : 8a 48 ad b3 12 ac b4 12  ce
29c8 : 20 93 e9 20 0a 17 20 22  4e
29d0 : 00 a0 00 b9 b7 12 20 99  17
29d8 : 14 c8 cc b6 12 90 f4 a9  27
29e0 : 22 20 99 14 20 9a e0 68  26
29e8 : aa 60 a0 13 b9 b3 12 99  02
29f0 : 00 01 88 10 f7 60 a0 13  c0
29f8 : b9 00 01 99 b3 12 88 10  33
2a00 : f7 60 8a 48 20 6b ea 20  1c
2a08 : 18 14 a0 13 20 24 14 99  5c
2a10 : b3 12 88 10 f7 68 aa 60  1f
2a18 : 8a 48 20 6b ea 20 30 14  d4
2a20 : a0 13 b9 b3 12 20 3c 14  6a
2a28 : 88 10 f7 68 aa 60 ad b6  95
2a30 : 12 18 69 04 8d a9 e5 8a  fc
2a38 : 48 20 6b ea 20 18 14 68  ac
2a40 : aa a0 03 20 24 14 d9 b3  b1
2a48 : 12 d0 08 c8 cc a9 e5 90  b0
2a50 : f2 f0 17 a0 00 20 24 14  4e
2a58 : 18 6d a1 e5 8d a1 e5 c8  5b
2a60 : 20 24 14 6d a2 e5 8d a2  1a
2a68 : e5 c8 60 0a 0a 8d a8 e5  86
2a70 : 0a 0a 6d a8 e5 6d a3 e5  14
2a78 : 85 26 ad a4 e5 69 00 85  c5
2a80 : 27 a9 26 a2 01 60 ce a5  f3
2a88 : e5 ec a5 e5 b0 0b e8 20  51
2a90 : 02 ea ca 20 18 ea e8 d0  dc
2a98 : f0 60 ad 98 e5 8d a1 e5  54
2aa0 : ad 99 e5 8d a2 e5 a2 00  29
2aa8 : ec a5 e5 b0 08 20 2e ea  06
2ab0 : f0 05 e8 d0 f3 a2 ff 60  8c
2ab8 : ad 9a e5 85 5e ad 9b e5  6a
2ac0 : 85 5f a0 01 20 2c eb 8d  6b
2ac8 : 9a e5 8c 9b e5 60 a4 2d  3a
2ad0 : a5 2e a2 00 84 5e 85 5f  45
2ad8 : 8a 48 ad a1 e5 85 5a ac  73
2ae0 : a2 e5 84 5b 18 6d b3 12  e1
2ae8 : 85 5c 98 6d b4 12 85 5d  1c
2af0 : a2 01 68 a8 20 2c eb 60  16
2af8 : ad a1 e5 85 5e ac a2 e5  41
2b00 : 84 5f e8 ec a5 e5 f0 1d  93
2b08 : 18 6d b3 12 85 5a 98 6d  6e
2b10 : b4 12 85 5b ad 9a e5 85  ec
2b18 : 5c ad 9b e5 85 5d a2 01  be
2b20 : a0 01 20 2c eb 8d 9a e5  30
2b28 : 8c 9b e5 60 a5 5a c5 5c  04
2b30 : a5 5b e5 5d b0 15 a5 5c  ab
2b38 : d0 02 c6 5d c6 5c 20 96  64
2b40 : f5 a6 5e a4 5f e8 d0 01  37
2b48 : c8 8a 60 a5 5e a4 5f 60  6b
2b50 : 4c 9b ee a0 5f a2 eb a9  47
2b58 : 06 20 38 f2 4c 9e ee 01  52
2b60 : 00 d9 eb 46 52 41 4d 45  ff
2b68 : 00 01 00 e5 eb 48 4f 52  88
2b70 : 49 5a 00 01 00 fd eb 56  53
2b78 : 45 52 54 49 00 01 00 1d  67
2b80 : ec 4d 45 4e 55 45 00 01  af
2b88 : 00 8e ec 50 55 52 47 45  a4
2b90 : 41 4c 4c 00 01 00 7c ec  e6
2b98 : 50 55 52 47 45 00 20 f9  d9
2ba0 : f0 86 dc 20 ed f0 86 db  47
2ba8 : 20 ed f0 86 dd 20 ed f0  44
2bb0 : 86 da 60 a5 dc c5 dd b0  45
2bb8 : 5c a5 dd c9 28 24 d7 10  ba
2bc0 : 02 c9 50 b0 50 60 a5 db  27
2bc8 : c5 da b0 49 a5 da c9 19  da
2bd0 : b0 43 a5 d3 29 10 85 d9  e2
2bd8 : 60 20 9e eb 20 b3 eb 20  fd
2be0 : c6 eb 4c de ed 20 f9 f0  34
2be8 : 86 da 20 ed f0 86 dc 20  98
2bf0 : ed f0 86 dd 20 b3 eb 20  42
2bf8 : cc eb 4c 4f ee 20 f9 f0  71
2c00 : 86 dd 20 ed f0 86 db 20  2d
2c08 : ed f0 86 da 20 c6 eb 20  92
2c10 : b9 eb 4c 75 ee 4c b0 f0  77
2c18 : a2 1a 4c be f0 ae a0 e5  85
2c20 : e8 f0 f5 20 9e eb a5 dc  9b
2c28 : 18 69 01 b0 e8 c5 dd b0  e1
2c30 : e4 20 b9 eb a5 db 16 69  7c
2c38 : 01 b0 da c5 da b0 d6 20  cf
2c40 : cc eb a5 dd 38 e5 dc a8  9e
2c48 : c8 98 0a 85 24 a5 da 38  db
2c50 : e5 db aa e8 ad 9c e5 ac  9b
2c58 : 9d e5 38 e5 24 b0 01 88  8f
2c60 : ca d0 f7 38 e9 36 b0 01  ac
2c68 : 88 cd 9a e5 98 ed 9b e5  6d
2c70 : 90 07 20 94 ec ee a0 e5  b3
2c78 : 60 4c b3 f0 ad a0 e5 f0  63
2c80 : 06 ce a0 e5 20 c7 ec 60  87
2c88 : ce a0 e5 20 c7 ec ad a0  00
2c90 : e5 d0 f5 60 ad 9c e5 85  c9
2c98 : 24 ad 9d e5 85 25 a9 24  27
2ca0 : a2 01 20 30 14 20 e9 ec  95
2ca8 : a2 03 b5 da 95 e4 ca 10  60
2cb0 : f9 20 19 ed 20 de ed e6  3c
2cb8 : e6 e6 e5 c6 e7 c6 e4 20  ec
2cc0 : 96 13 42 c1 4c de ec ad  73
2cc8 : 9c e5 85 24 ad 9d e5 85  a7
2cd0 : 25 a9 24 a2 01 20 18 14  c1
2cd8 : 20 40 ed 20 03 ed a5 24  16
2ce0 : 8d 9c e5 a5 25 8d 9d e5  ea
2ce8 : 60 a2 00 b5 e0 20 77 ed  19
2cf0 : e8 e0 0d 90 f6 a2 00 bd  9e
2cf8 : 54 03 20 77 ed e8 e0 0e  8b
2d00 : 90 f5 60 a2 0e 20 6b ed  63
2d08 : ca 9d 54 03 d0 f7 a2 0d  88
2d10 : 20 6b ed ca 95 e0 d0 f8  50
2d18 : 60 a4 e5 84 eb a6 e6 86  71
2d20 : ec 20 40 17 20 77 ed a5  d0
2d28 : f2 20 77 ed e6 ec a5 e7  02
2d30 : c5 ec b0 ed e6 eb a5 e4  83
2d38 : c5 eb b0 e1 20 e9 ec 60  21
2d40 : 20 03 ed a4 e4 84 eb a6  61
2d48 : e7 86 ec 20 6b ed 85 f2  d4
2d50 : 20 6b ed 20 4d 17 c6 ec  28
2d58 : 30 06 a5 ec c5 e6 b0 eb  c0
```

Listing 4. Der zweite Teil von »MASTER« (Schluß)

```
2d60 : c6 eb 30 06 a5 eb c5 e5  85
2d68 : b0 dd 60 a0 00 20 24 14  ed
2d70 : e6 24 d0 02 e6 25 60 a4  3f
2d78 : 24 d0 02 c6 25 c6 24 a0  b8
2d80 : 00 4c 3c 14 a0 0f 8e b3  5c
2d88 : 12 8c b4 12 20 40 17 a2  f5
2d90 : 0f a4 d9 d0 0a dd be ed  e8
2d98 : f0 0d ca d0 f8 f0 08 dd  ce
2da0 : ce ed f0 03 ca d0 f8 8a  2d
2da8 : 0d b3 12 2d b4 12 05 d9  5d
2db0 : f0 ed aa bd be ed a4 f1  cb
2db8 : 84 f2 20 4d 17 60 20 40  dd
2dc0 : 5d 70 40 40 6e 72 5d 6d  38
2dc8 : 5d 6b 7d 71 73 5b 20 66  c8
2dd0 : 61 68 66 66 6a 69 61 62  08
2dd8 : 61 65 64 63 67 6b a5 ec  b4
2de0 : 48 a5 eb 48 a6 dc 86 ec  44
2de8 : a4 db 84 eb a2 03 20 84  e4
2df0 : ed e6 ec a5 ec c5 dd b0  16
2df8 : 0a a2 05 a0 0d 20 86 ed  71
2e00 : 4c f1 ed a2 06 20 84 ed  64
2e08 : e6 eb a5 eb c5 da b0 0a  d5
2e10 : a2 0a a0 0b 20 86 ed 4c  c7
2e18 : 08 ee a2 0c 20 84 ed c6  2d
2e20 : ec a5 dc c5 ec b0 0a a2  90
2e28 : 05 a0 07 20 86 ed 4c 1f  8a
2e30 : ee a2 09 20 84 ed c6 eb  60
2e38 : a5 db c5 eb b0 0a a2 0a  b4
2e40 : a0 0e 20 86 ed 4c 36 ee  b8
2e48 : 68 85 eb 68 85 ec 60 a5  07
2e50 : ec 48 a5 eb 48 a6 dc 86  81
2e58 : ec a4 da 84 eb a2 01 2c  0e
2e60 : a2 05 20 84 ed e6 ec a5  32
2e68 : ec c5 dd 90 f3 a2 04 20  65
2e70 : 84 ed 4c 48 ee a5 ec 48  67
2e78 : a5 eb 48 a6 dd 86 ec a4  09
2e80 : db 84 eb a2 02 2c a2 0a  0d
2e88 : 20 84 ed e6 eb a5 eb c5  6a
2e90 : da 90 f3 a2 08 20 84 ed  73
2e98 : 4c 48 ee 4c ff ef a0 a7  9f
2ea0 : a2 ee a9 05 4c 38 f2 01  19
2ea8 : 00 dc ee 4d 45 4d 44 45  d6
2eb0 : 46 00 01 00 42 ef 49 4e  9c
2eb8 : 53 45 52 54 00 00 00 0d  e7
2ec0 : ef 4d 45 4d 24 00 00 00  93
2ec8 : 81 ef 42 41 53 49 43 24  ce
2ed0 : 00 00 00 c0 ef 42 41 53  a5
2ed8 : 4c 4e 24 00 20 db f0 48  8a
2ee0 : 98 48 20 a5 f0 20 11 f1  91
2ee8 : a8 68 85 16 68 85 17 a9  4b
2ef0 : 24 a2 01 20 18 14 a9 16  9f
2ef8 : ae d5 03 20 30 14 98 a8  ad
2f00 : f0 0a 88 20 24 14 20 b3  f7
2f08 : 14 98 d0 f6 60 20 ea f4  18
2f10 : 20 ed f0 8a a8 a9 16 ae  42
2f18 : d5 03 20 18 14 98 48 20  e1
2f20 : 23 f1 a9 64 a2 01 20 30  46
2f28 : 14 68 20 ff ee 4c 29 f1  52
2f30 : 20 17 f1 a8 a9 00 99 00  6e
2f38 : 0c 98 f0 05 48 20 ca f2  04
2f40 : 68 60 20 96 13 d9 84 20  06
2f48 : 11 f1 20 33 ef a5 7d c9  76
2f50 : ff d0 29 a5 7e c9 09 d0  b2
2f58 : 23 20 61 17 20 da 16 b0  59
2f60 : 18 20 96 13 a0 50 a5 3b  2a
2f68 : c5 16 a5 3c e5 17 b0 0c  1b
2f70 : a9 4d 48 a9 e4 48 4c dd  85
2f78 : 02 4c ad f0 a2 1a 4c be  d3
2f80 : f0 20 30 ef f0 26 a9 00  71
2f88 : 85 61 a9 0c 85 62 a5 4b  42
2f90 : 48 a5 4c 48 20 cb 17 a0  c5
2f98 : 00 84 11 20 96 13 49 51  ec
2fa0 : 20 e2 17 68 85 4c 68 85  6c
2fa8 : 4b ad 60 17 48 a9 00 85  a2
2fb0 : 16 a9 0d 85 17 a9 16 a2  eb
2fb8 : 00 20 18 14 68 4c 1e ef  92
2fc0 : 20 ea f4 a9 16 ae d5 03  fc
2fc8 : 20 18 14 a0 00 20 24 14  c7
2fd0 : 99 00 0c c8 d0 f7 a9 00  f9
2fd8 : 85 61 a9 0c 85 62 a5 4b  92
2fe0 : 48 a5 4c 48 20 cb 17 ae  31
2fe8 : 02 0c ad 03 0c 20 96 13  fe
2ff0 : 23 51 4c a0 ef 28 43 29  83
2ff8 : 48 4a 42 01 0b 56 00 60  3a
```

```
100 PRINT                                   <T3U>
110 PRINT "***{3SPACE}MASTER{3SPACE}***"    <QBC>
120 PRINT "(C)HJB{4SPACE}01.11.86           <4UB>
130 BANK 0                                  <SFR>
140 BOOT "/BUTLER.1"                        <VLK>
150 BOOT "/BUTLER.2"                        <V9N>
160 BOOT "/MASTER.1"                        <72L>
170 BOOT "/MASTER.2"                        <7UM>
180 NEW                                     <O32>
```

Listing 5. Das Programm »MASTER.LADER« sorgt für den Aufruf und die Initialisierung von »MASTER« und »Butler«

Basic »de Luxe« auf engstem Raum

Dieses kurze Programm erweitert das ohnehin schon komfortable Basic 7.0 um etliche sehr hilfreiche Befehle. Außerdem erfahren Sie, wie man mit geringem Aufwand eigene Befehle in die Interpreterschleife des C 128 einbinden kann.

Der C 128 besitzt das bislang leistungsfähigste Basic aller Commodore-Computer, doch entdeckt man bald einige Mängel, die der anfänglichen Programmier-Begeisterung gewisse Grenzen setzen. So mancher Basic-Programmierer und C 128-Besitzer hegt deshalb den Wunsch, den Befehlsumfang seines Computers zu erweitern. Oftmals scheitern die Bemühungen an ungenügenden Kenntnissen über den Interpreter des C 128.

Daß es jedoch auf einfachste Weise möglich ist, eigene Befehle und Funktionen in die Interpreterschleife des C 128 einzubinden, zeigt die Erweiterung »xBasic 128«, die dem Anwender 52 neue und interessante Befehle zur Verfügung stellt. Wir werden sie sogleich in der anschließenden Befehlsbeschreibung vorstellen. Alle Anweisungen, die in der Form »v=BEFEHL(x)« aufgeführt sind, stellen Funktionen dar, das heißt, sie liefern einen Zahlenwert, der entweder mit PRINT ausgegeben oder einer Variablen zugewiesen werden muß.

Sound-Befehle

Die folgenden Befehle wirken auf alle mit PLAY oder SOUND erzeugten Klänge.

SSOUND
läßt den gerade gespielten Ton ausklingen.

CSOUND
bricht den gerade gespielten Sound ab. Dabei verschwindet auch das bisweilen lästige Brummen im Hintergrund.

RSOUND
kompletter Sound-Reset. Die nach dem Einschalten vorgegebenen Werte, zum Beispiel für die Hüllkurven, werden wiederhergestellt.

Grafisch in Höchstform

SPRINV n (n: 1 bis 8)
invertiert das Sprite mit der Nummer »n«.

SPRPOINT n,w (n: 1 bis 8, w: 0 bis 255)
setzt den Sprite-Pointer für Sprite »n« auf den Wert »w«. Damit lassen sich Sprites animieren, indem ihnen verschiedene Bit-Muster zugeordnet werden.

v=RSPOINT (n) (n: 1 bis 8)
liefert den momentanen Wert des Pointers für Sprite »n«.

v=SBUMP (n,m) (n und m: 1 bis 8)
Der normale BUMP-Befehl liefert leider nur den gleichen Wert wie das Kollisions-Register des VIC-Chips, in dem für jedes Sprite ein Bit reserviert ist. Mit SBUMP läßt sich nun gezielt prüfen, ob seit der letzten Abfrage Sprite »n« mit Sprite »m« kollidiert ist. Dann liefert die Funktion den Wert 1, ansonsten 0. Eine Abfrage setzt das Bump-Flag des Interpreters auf 0 zurück.

v=DBUMP(n) (n: 1 bis 8)
Dieser Befehl ähnelt SBUMP, nur wird hier geprüft, ob Sprite »n« mit Bildschirmdaten kollidiert ist.

Für die folgenden Befehle, mit Ausnahme der Anweisung CHARSET, muß die Grafik aktiviert sein. Andernfalls wird deren Anwendung mit einem »NO GRAPHICS AREA ERROR« quittiert, damit ein im Speicher befindliches Basic-Programm nicht versehentlich zerstört wird.

INVERT
invertiert die Hires-Grafik. Im Multicolor-Modus bewirkt diese Anweisung lediglich ein Vertauschen der Farben.

SETCOLOR
Farbeinstellungen, die mit dem COLOR-Befehl vorgenommen werden, wirken normalerweise nicht auf eine bereits fertige Grafik, sondern erst bei Löschen des Bildschirms. SETCOLOR setzt die neuen Farben, ohne die Grafik zu zerstören.

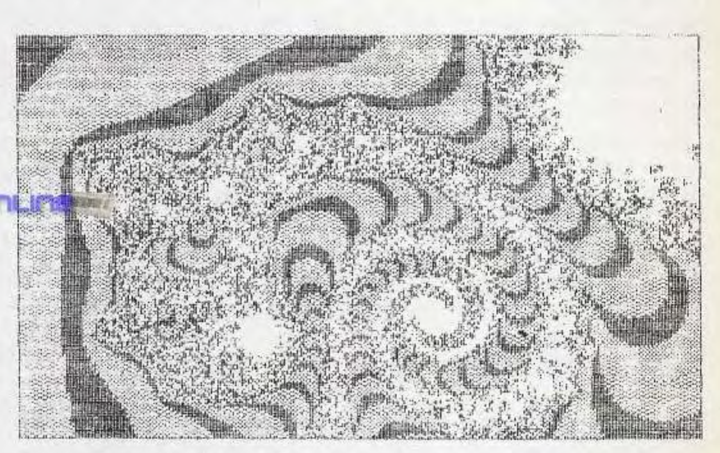


Bild 1 Eine Multicolor-Grafik wurde zunächst mit GREY umgewandelt und mit HCOPY gedruckt

GREY
wandelt eine Multicolor-Grafik in eine Hires-Grafik mit verschiedenen Grautönen um. Falls nach Umschalten in den Hires-Modus nichts auf dem Bildschirm zu sehen ist, müssen Sie einmal den SETCOLOR-Befehl anwenden. Die Umwandlung läßt sich jedoch nicht wieder rückgängig machen!

HCOPY
gibt eine Hardcopy der 40-Zeichen-Grafik auf einem MPS 801-Drucker und Kompatiblen aus. In Bild 1 sehen Sie einen Beispielausdruck.

CHARSET n (n: 0 oder 1)
Im Grafik-Modus steht für den CHAR-Befehl normalerweise nur der Groß/Grafik-Zeichensatz zur Verfügung. CHARSET 1 schaltet den Klein/Groß-Zeichensatz ein. CHARSET 0 schaltet zurück in den Groß-Grafik-Modus.

Die folgenden Befehle erleichtern den Umgang mit dem 80-Zeichen-Videochip VDC.

VSET n,w (n: 0 bis 36, w: 0 bis 255)
schreibt den Wert »w« in das Register »n« des VDC-Chips.

v=VREG(n) (n: 0 bis 36)
liefert den Inhalt des VDC-Registers »n«.

VPOKE a,w1[,w2,w3,...] (a: 0 bis 65535, wx: 0 bis 255)
schreibt den Wert »w1« in die Adresse »a« des VDC-RAMs. Da das VDC-RAM nur 16 KByte groß ist, sind für »a« nur Werte bis 16383 sinnvoll, mit »a=16384« wird wieder peicherstelle 0 angesprochen. Dieser Befehl nutzt die Eigenschaft des VDC, das Adreß-Register selbständig zu erhöhen, so daß hinter »w1« beliebig viele weitere Werte angegeben werden können, die in die auf »a« folgenden Adressen geschrieben werden. Dies ist besonders für Änderungen des Zeichensatzes nützlich.

v=VPEEK(a) (a: 0 bis 65535)
liefert den Inhalt der Adresse »a« des VDC-RAMs. Für die Werte von »a« gilt das unter VPOKE Gesagte.

Mathematische Leckerbissen

v=COT(x)
liefert den Cotangens von »x«, der Wert »x« ist im Bogenmaß anzugeben.

v=DLOG(x)
liefert den dekadischen Logarithmus von »x«. Der Wert »x« muß größer als 0 sein.

v=RAD(x)
liefert das Bogenmaß von »x«. Der Wert »x« ist als Gradzahl anzugeben.

v=DEG(x)
liefert das Gradmaß von »x«. Der Wert »x« ist im Bogenmaß anzugeben.

v=DIV(x,y)
liefert den ganzzahligen Teil der Division »x/y«.

v=MOD(x,y)
liefert den Rest der Division »x/y«.

v=LOW(x)
liefert das niederwertige Byte der 16-Bit-Integerzahl »x«.

v=HIGH(x)
liefert analog zu LOW das höherwertige Byte.

Speichermanipulationen wie mit TEDMON

SETBOT b,a (b: 0 oder 1, a: 0 bis 65535)
setzt die Untergrenze des für Basic verfügbaren Speichers in Bank »b« auf die Adresse »a«.

SETTOP b,a (b: 0 oder 1, a: 0 bis 65535)
setzt analog zu SETBOT die Obergrenze. Falls Sie mit diesen Befehlen im Programm-Modus den Variablenspeicher (Bank 1) manipulieren, sollten Sie diesen ein CLR folgen lassen, um ein Chaos bei der Verarbeitung der Variablen zu vermeiden.

v=RBOT(b) (b: 0 oder 1)
liefert die aktuelle Untergrenze des für Basic in Bank »b« zur Verfügung stehenden Speichers.

v=RTOP(b) (b: 0 oder 1)
liefert analog zu RBOT die Obergrenze.

MOVE b1,a,e,b2,z (b1, b2: 0 bis 15; a, e, z: 0 bis 65535)
verschiebt den Speicherinhalt von Adresse »a« bis Adresse »e« in Bank »b1« an die Adresse »z« in Bank »b2«.

Die folgenden Befehle wirken jeweils auf die mit dem BANK-Befehl eingestellte Speicherbank.

DOKE a,w (a und w: 0 bis 65535)
schreibt die 16-Bit-Integerzahl »w« in Low/Byte-Format an die Adressen »a« und »a+1«.

v=DEEK(a) (a: 0 bis 65535)
interpretiert den Inhalt der Adressen »a« und »a+1« als 16-Bit-Integerzahl im Low/High-Format und liefert deren Wert.

FILL a,e,w (a und e: 0 bis 65535, w: 0 bis 255)
füllt den Speicher von Adresse »a« bis Adresse »e« mit dem Wert »w«.

Ladebefehle und Programmsprünge

Die folgenden zwei Befehle dienen dazu, ein Basicprogramm an ein bereits im Speicher befindliches anzuhängen.

MERGE "<filename>"[,g]
Dieser MERGE-Befehl wird wie der LOAD-Befehl angewendet. Somit ist »g« die Gerätenummer für ein Disketten-Laufwerk oder die Datasette. Dieser Parameter muß nicht angegeben werden, voreingestellt ist hier Gerätenummer 1. Wird auch kein Filename angegeben, wird das nächste Programm von Kassette geMERGEt. Ein absolutes Laden durch Anhängen von »,1« hinter der Gerätenummer ist zwar möglich, aber wohl in den wenigsten Fällen sinnvoll.

DMERGE "<filename>"[,D lw][, U g]
Die Parameter dieses Befehls entsprechen denen des DLOAD-Befehls. Hinter dem File-Namen kann also mit »D« die Laufwerksnummer (lw; 0 oder 1) angegeben werden, mit »U« die Gerätenummer (g).

Für die File-Namen sind die üblichen Abkürzungen erlaubt.

VGOTO x und **VGODUB x**
Diese Befehle entsprechen den bekannten Anweisungen GOTO und GOSUB. Die Sprungadresse kann aber hier auch in einer Variablen übergeben werden, also etwa:

```
10 x=100:vgoto x
```


Sonstige Befehle und Funktionen

SCNSWAP
schaltet vom 40- auf den 80-Zeichen-Bildschirm und zurück.

FULLSCREEN
hebt jedes bestehende Bildschirmfenster auf. Die Cursor-Position wird nicht beeinflußt.

FKEY [n,"<string>"] (n: 1 bis 10)
entspricht im wesentlichen dem bekannten KEY-Befehl. Mit FKEY sind jedoch alle Tasten, die belegbar sind, auch ansprechbar. Hierbei entspricht »n=9« der RUN/STOP-Taste und »n=10« der HELP-Taste. FKEY ohne Parameter gibt die Belegung aller 10 Tasten auf dem Bildschirm aus.

FKEY
stellt die nach dem Einschalten vorhandene Belegung aller 10 Funktionstasten wieder her.

REPEAT ON/OFF
Mit REPEAT OFF wird die automatische Tastaturwiederholung ausgeschaltet. REPEAT ON schaltet sie wieder ein.

CURSOR
Dieser Befehl kann in verschiedenen Formen angewendet werden. Für den 80-Zeichen-Bildschirm gilt:
CURSOR FAST: schnell blinkender Cursor
CURSOR BLOCK: Block-Cursor
CURSOR OFF: Cursor verschwindet völlig
CURSOR SLOW: normaler, langsam blinkender Cursor

Auf dem 40-Zeichen-Bildschirm schaltet CURSOR BLOCK den Block-Cursor ein, CURSOR OFF den normalen Cursor.

OLD
stellt ein durch NEW oder eines Reset gelöschten Basic-Programm wieder her. Nach einem Reset muß zunächst durch SYS 4864 die Basic-Erweiterung wieder aktiviert werden. War zuvor die Grafik eingeschaltet, müssen Sie

vor Anwendung des OLD-Befehls zunächst »GRAPHIC 1: GRAPHIC 0« eingeben.

PAUSE
hält den Programmablauf an, bis eine Taste gedrückt wird.

SETUSR a (a: 0 bis 65535)
setzt den USR-Vektor auf die Adresse »a«.

Befehl	Abkürzung	Befehl	Abkürzung
chr$	chR	dopen	doP
def	keine	filter	filT
input	inP	graphic	grA
load	loA	movspr	movS
print#	prl	record	reC
read	reA	rsppos	rsP
char	keine	scnclr	scnC
concat	coN	sprite	spriT
bload	bloA	xor	keine
dload	dloA		

Tabelle 1. Die neuen Befehle führen zu einer Änderung einiger Basic-Abkürzungen. Die Tabelle zeigt die davon betroffenen Anweisungen

$1300	INIT	Initialisierung der Basic-Erweiterung
$135d	CRUNCH	Neuen Befehl in Token umwandeln
$138f	LIST	Neuen Befehl listen
$13aa	EXEC	Neuen Befehl ausführen
$13d0	COMADR	Liste der Adressen (-1) der neuen Befehle
$1414	FNADR	Liste der Adressen der neuen Funktionen
$143c	CMDLST	Liste der Schlüsselwörter (Befehle)
$14f2	FNLST	Liste der Schlüsselwörter (Funktionen)
$1548	CSOUND	Hier beginnen die Routinen zu den neuen Anweisungen
$1b57	HCERR	(Hier kann eine eigene Hardcopy-Routine beginnen) Ausgabe von »DEVICE NOT PRESENT ERROR«, wenn bei HCOPY der Drucker nicht eingeschaltet ist
$1b5f	HCOPY	Beginn der eigentlichen Hardcopy-Routine
xBasic 128 benutzt außerdem die Adressen $77/78 und $fa bis $fe als Zwischenspeicher sowie den Bereich ab $100 (normalerweise für Filenamen reserviert) als Bytepuffer für HCOPY.		

Tabelle 2. Die wichtigsten Adressen von xBasic 128

Bei den folgenden Funktionen ist der Parameter »d« jeweils eine Dummy-Variable, die zwar angegeben werden muß, deren Wert jedoch völlig gleichgültig ist.

v=PRGLEN(d)
liefert die Länge des momentan im Speicher befindlichen Basic-Programms einschließlich der beiden abschließenden Null-Bytes, also die eigentliche Programmlänge plus 2.

v=DLINE(d)
liefert die Zeilennummer der aktuellen DATA-Zeile.

v=LDEND(d)
liefert die Adresse, bis zu der beim letzten Ladevorgang geladen wurde. Damit läßt sich beispielsweise die Länge eines mit BLOAD geladenen Binär-Files ermitteln, sofern man dessen Anfangsadresse kennt.

v=RBANK(d)
liefert die Nummer der momentan mit dem BANK-Befehl eingestellten Speicherbank.

Zum Abschalten von xBasic 128 existieren zwei Möglichkeiten:

OFF
schaltet die Basic-Erweiterung ab. Gleichzeitig wird ein kurzes Maschinenprogramm in die NMI-Routine eingebunden, so daß xBasic 128 durch einen Druck auf die RESTORE-Taste wieder aktiviert werden kann. Der NMI-Vektor enthält danach wieder seinen ursprünglichen Wert.

XOFF
schaltet die Befehlserweiterung völlig ab.

Verwendet man XOFF in einem Programm, läßt sich bei zeitkritischen Routinen die Arbeitsgeschwindigkeit des Interpreters erhöhen. Anschließend darf xBasic 128 wieder mit SYS 4864 gestartet werden.

Trotz dieser enormen Leistungsfähigkeit belegt xBasic 128 nur etwa 2 KByte im Speicher, da verschiedene ROM-Routinen des C128 genutzt werden. Der OLD-Befehl ist beispielsweise durch ein nur 8 Bytes langes Unterprogramm realisiert.

Das komplette Programm »xBasic 128« finden Sie in Listing 1. Es hat den Namen »xbasic128.obj« und muß im C64-Modus mit dem MSE eingegeben und gespeichert werden. Von nun an kann man »xBasic 128« mit

```
BOOT "xbasic128.obj", ON B 15
```

starten, um in den Genuß dieser interessanten Basic-Erweiterung zu kommen. Doch sei zuvor auf einige Besonderheiten hingewiesen.

Beachtenswertes

Alle Befehle von xBasic 128 dürfen nach den üblichen Regeln abgekürzt werden. Da sie jedoch vom Basic-Interpreter mit Vorrang behandelt werden, ergeben sich für die Abkürzungen der Standard-Basic-Befehle einige Änderungen, die wir in einer Tabelle 1 zusammengestellt haben (die Befehle und Abkürzungen sind dabei im Klein-/Großschrift-Modus angegeben). Nicht aufgeführte Basic-Kommandos werden weiterhin wie gewohnt abgekürzt.

Daneben sollte die Standard-Belegung der RUN/STOP-Taste mit dem neuen Befehl FKEY folgendermaßen geändert werden, denn das ursprüngliche Kürzel für DLOAD (»dL«) führt aufgrund des neuen Befehls DLOG zu einem »SYNTAX ERROR«:

```
fkey 9,"dloA"+chr$(34)+"*"+chr$(13)+"run"+chr$(13)
```

Die Hardcopy-Routine, die mit HCOPY aktiviert wird, läßt nur den Ausdruck auf einem MPS 801-Drucker und Kompatiblen zu. Besitzer von anderen Druckern seien jedoch getröstet, denn diese Routine befindet sich am Ende der Erweiterung ab Adresse $1b57 und kann somit problemlos durch eine an den eigenen Drucker angepaßte Version ersetzt werden. Der dokumentierte Quelltext der Hardcopy-Routine in Listing 2 wird dabei wertvolle Hilfe leisten. Interessant ist, daß diese Routine keine absoluten Sprünge enthält und damit in jedem Speicherbereich in Bank 0 lauffähig ist, der nicht unter dem ROM liegt.

Für Assembler-Profis: Die »persönliche« Basic-Erweiterung

Wem die Befehle von xBasic 128 nicht genügen, der kann sich eine Basic-Erweiterung nach eigenen Wünschen zusammenstellen. Die Tabelle 2 gibt dabei Auskunft über wichtige Adressen und Routinen von xBasic 128. Das Hauptprogramm (die Interpreterschleife), dessen Quelltext im TOPASS-Format in Listing 3 abgedruckt ist, befindet sich ab Adresse $1300 bis $13c0. Diese Erweiterung ist universell anwendbar. Sie müssen »nur« die Routinen zu Ihren eigenen Befehlen hinzuschreiben. Als Beispiel in Listing 3 dienen hier die beiden Kommandos FULLSCREEN und TEN (ab Zeile 2030). Sie zeigen, wie die neuen Routinen an das Interpreter-Programm angepaßt werden.

(Michael Schmand/Michael Thomas)

```
Name : xbasic128.obj       1300 1c00
-----------------------------------
1300 : a2 05 bd 57 13 9d 0c 03   d3
1308 : ca 10 f7 a9 bc 8d fc 02   3d
1310 : a9 13 8d fd 02 20 81 92   b2
1318 : 93 0d 0e 00 24 d7 10 0a   0b
1320 : a2 13 a9 20 20 69 92 ca   e7
1328 : d0 f8 20 81 92 20 c3 31   48
1330 : 32 38 20 58 c2 41 53 49   a8
1338 : 43 20 d6 33 2e 30 20 20   cd
1340 : 20 28 c3 29 20 31 39 38   6b
1348 : 37 20 cd 2e 20 d3 43 48   07
1350 : 4d 41 4e 44 0d 00 60 5d   67
1358 : 13 8f 13 aa 13 48 a9 14   8f
1360 : a0 3c 20 e2 43 90 0d aa   c5
1368 : 68 8a 29 7f 18 69 27 a2   fe
1370 : 00 4c b2 43 a9 14 a0 f2   4f
1378 : 20 e2 43 b0 05 68 38 4c   fd
1380 : 21 43 aa 68 8a 29 7f 18   1b
1388 : 69 0b a2 ff 4c b2 43 09   99
1390 : 80 e8 f0 0b 38 e9 27 aa   e7
1398 : a9 14 a0 3c 4c 6a 51 38   c9
13a0 : e9 0b aa a9 14 a0 f2 4c   99
13a8 : 6a 51 29 7f 38 e9 27 0a   79
13b0 : a8 b9 d1 13 48 b9 d0 13   c7
13b8 : 48 4c 80 03 38 e9 0b 0a   ba
13c0 : a8 b9 15 14 85 58 b9 14   37
13c8 : 14 85 57 20 56 00 18 60   ff
13d0 : 47 15 55 15 5a 15 29 c0   0e
13d8 : 89 15 8f 15 23 ca b0 15   e8
13e0 : 07 16 2d 16 37 16 65 16   e6
13e8 : 9e 16 be 16 e4 16 f3 16   ff
13f0 : 2f 1a 26 17 3b 17 00 00   05
13f8 : 45 17 58 17 ec 17 f7 17   57
1400 : 19 18 6b 79 5a 18 1a 19   30
1408 : 39 19 55 19 0e 1a f7 1a   0c
1410 : 67 1a 5e 1b 61 15 69 15   0e
1418 : 9c 15 bd 15 d0 15 4d 16   68
1420 : 86 16 2f 17 10 17 d2 17   93
1428 : df 17 77 19 84 19 91 19   1d
1430 : b0 19 eb 19 c1 1a df 1a   2b
1438 : 18 1b 32 1b 43 53 4f 55   85
1440 : 4e c4 53 53 4f 55 4e c4   92
1448 : 52 53 4f 55 4e c4 53 43   a1
1450 : 4e 53 57 41 d0 56 47 4f   c1
1458 : 54 cf 56 47 4f 53 55 c2   7d
1460 : 46 55 4c 4c 53 43 52 45   11
1468 : 45 ce 52 4b 45 d9 44 4f   e5
1470 : 4b c5 53 45 54 55 53 d2   fe
1478 : 56 53 45 d4 56 50 4f 4b   1f
1480 : c5 53 45 54 43 4f 4c 4f   49
1488 : d2 53 50 52 49 4e d6 43   4b
1490 : 48 41 52 53 45 d4 53 50   61
1498 : 52 50 4f 49 4e d4 46 49   47
14a0 : 4c cc 4f 4c c4 4f 46 c6   1d
14a8 : a0 58 4f 46 c6 47 52 45   8b
14b0 : d9 50 41 55 53 c5 52 45   e4
14b8 : 50 45 41 d4 43 55 52 53   65
14c0 : 4f d2 42 4c 4f 43 cb 46   5d
14c8 : 4b 45 d9 49 4e 56 45 52   a7
14d0 : d4 4d 45 52 47 c5 44 4d   35
14d8 : 45 52 47 c5 53 45 54 54   2a
14e0 : 4f d0 53 45 54 42 4f d4   53
14e8 : 4d 4f 56 c5 48 43 4f 50   a8
14f0 : d9 00 4c 4f d7 48 49 47   3a
14f8 : c8 50 52 47 4c 45 ce 43   17
1500 : 4f d4 44 45 45 cb 56 52   24
1508 : 45 c7 56 50 45 45 cb 44   07
1510 : 4c 49 4e c5 52 53 50 4f   ed
1518 : 49 4e d4 4c 44 45 4e c4   78
1520 : 52 42 41 4e cb 52 41 c4   8b
1528 : 44 45 c7 44 4c 4f c7 53   8e
1530 : 42 55 4d d0 44 42 55 4d   d0
1538 : d0 52 54 4f d0 52 42 4f   78
1540 : d4 44 49 d6 4d 4f c4 00   c6
1548 : 20 7a 41 a9 00 8d 04 d4   51
1550 : 8d 0b d4 8d 12 d4 a2 ff   9c
1558 : 4c 55 41 20 7a 41 4c 12   aa
1560 : 41 20 72 15 a9 00 4c 03   c2
1568 : af 20 72 15 a8 a9 00 4c   d7
1570 : 03 af 20 56 79 20 da 77   11
1578 : a5 63 c9 91 b0 09 20 c7   d7
1580 : 8c a4 67 a5 66 60 ea 4c   9b
1588 : 28 7d 20 12 88 4c e2 59   e2
1590 : 20 1d 5a 20 86 03 20 8a   f0
1598 : 15 4c f6 4a 20 56 79 20   b5
15a0 : da 77 ad 10 12 38 e5 2d   78
15a8 : a8 ad 11 12 e5 2e 4c c9   42
15b0 : 84 a2 4c bd a8 ce 9d 00   c8
15b8 : 10 ca 10 f7 60 20 56 79   84
15c0 : 20 da 77 20 59 94 a9 1b   46
15c8 : a0 95 20 89 8a 4c 4c 8b   bf
15d0 : 20 56 79 a5 17 48 a5 16   a5
15d8 : 48 20 da 77 20 15 88 ae   00
15e0 : d5 03 a0 00 a9 16 20 74   14
15e8 : ff 85 77 e6 16 d0 02 e6   22
15f0 : 17 ae d5 03 a0 00 a9 16   11
15f8 : 20 74 ff a8 68 85 16 68   43
1600 : 85 17 98 a4 77 4c c9 84   d5
1608 : 20 12 88 85 78 84 77 20   ce
1610 : 0f 88 48 98 a0 00 a2 77   0c
1618 : 8e b9 02 ae d5 03 20 77   be
1620 : ff 68 e6 77 d0 02 e6 78   a5
1628 : ae d5 03 4c 77 ff 20 12   27
1630 : 88 8c 19 12 8d 1a 12 60   3a
1638 : 20 f4 87 e0 25 b0 24 86   46
1640 : 77 20 09 88 8a a6 77 20   17
1648 : 45 a8 4c cc cd 20 56 79   b8
1650 : 20 f7 87 e0 25 b0 0c 20   b2
1658 : 45 a8 20 da cd a8 a9 00   1d
1660 : 4c 03 af 4c 28 7d 20 12   b6
1668 : 88 a2 12 20 45 a8 20 cc   7e
1670 : cd e8 98 20 cc cd 20 1e   d3
1678 : 9e 90 0a 20 45 a8 8a 20   e9
1680 : ca cd 4c 76 16 60 20 72   dd
1688 : 15 a2 12 20 45 a8 20 cc   2b
1690 : cd e8 98 20 cc cd 20 d8   69
1698 : cd a8 a9 00 4c 03 af 20   00
16a0 : 74 a0 20 5c 6a 24 d8 10   43
16a8 : 03 20 17 6b 4c 39 6b 00   2b
16b0 : 0e 40 0e 80 0e c0 0e 00   91
16b8 : 0f 40 0f 80 0f c0 0f 20   2f
16c0 : f4 87 ca 30 9e e0 08 b0   a3
16c8 : 9a 8a 0a aa bd af 16 85   3c
16d0 : 77 e8 bd af 16 85 78 a0   d1
16d8 : 00 b1 77 49 ff 91 77 c8   b4
16e0 : c0 40 d0 f5 60 20 f4 87   9d
16e8 : 8a f0 03 a9 d8 2c a9 d0   18
16f0 : 8d ec 11 60 20 f4 87 ca   a1
16f8 : 30 13 e0 08 b0 0f 86 77   77
1700 : 20 09 88 8a a6 77 9d f8   a7
1708 : 07 9d f8 1f 60 4c 28 7d   04
1710 : 20 72 15 c9 00 d0 f6 88   5b
1718 : 30 f3 c0 08 b0 ef b9 f8   d6
1720 : 07 a8 a9 00 4c 03 af a9   d5
1728 : 01 a8 91 2d 4c e5 5e 20   35
1730 : 56 79 20 da 77 a4 41 a5   93
1738 : 42 4c c9 84 a9 49 8d 18   ee
1740 : 03 a9 17 8d 19 03 4c 51   0d
1748 : 42 20 00 13 a2 40 a0 fa   a1
1750 : 8e 18 03 8c 19 03 4c 40   98
1758 : fa 20 74 a0 a9 20 85 fb   3d
1760 : a9 00 85 fa 20 75 17 e6   a2
1768 : fa d0 f9 a4 fb c8 c0 40   67
1770 : 84 fb d0 f0 60 a2 00 a9   b2
1778 : 03 85 fc a0 00 b1 fa 25   55
1780 : fc c5 fc f0 14 b1 fa 25   c1
1788 : fc f0 0e dd ce 17 d0 06   31
1790 : 20 a5 17 4c 99 17 20 b2   0a
1798 : 17 e8 e0 04 f0 06 06 fc   2d
17a0 : 06 fc d0 d9 60 a5 fa 29   05
17a8 : 01 f0 06 b1 fa 45 fc 91   ca
17b0 : fa 60 a5 fa 29 01 d0 0b   97
17b8 : e0 03 f0 f5 e0 01 f0 f1   d2
17c0 : 4c ab 17 e0 02 f0 ea e0   d9
17c8 : 00 f0 e6 4c ab 17 01 04   03
17d0 : 10 40 20 56 79 20 da 77   c6
17d8 : a5 af a4 ae 4c c9 84 20   b9
17e0 : 56 79 20 da 77 ac d5 03   90
17e8 : a9 00 4c c9 84 a9 80 8d   90
17f0 : 21 0a 2c 21 0a 30 fb 60   18
17f8 : c9 91 d0 08 a9 80 8d 22   d8
1800 : 0a 4c 80 03 c9 fe d0 0f   a7
1808 : 20 80 03 c9 39 d0 08 a9   f0
1810 : 40 8d 22 0a 4c 80 03 4c   4e
1818 : 6c 79 c9 fe d0 f9 20 80   f1
1820 : 03 c9 39 d0 0d a9 20 8d   2a
1828 : 2b 0a a9 00 8d 26 0a 4c   8e
1830 : 80 03 c9 25 d0 08 a9 40   bd
1838 : 8d 2b 0a 4c 80 03 c9 26   fb
1840 : d0 08 a9 60 8d 2b 0a 4c   7e
1848 : 80 03 c9 40 d0 c9 a9 80   c7
1850 : 8d 2b 0a a9 40 8d 26 0a   48
1858 : 4c 80 03 f0 23 c9 91 f0   6c
1860 : 0e c9 fe d0 0d 20 80 03   06
1868 : c9 39 f0 03 4c 6c 79 4c   11
1870 : 46 48 20 f4 87 ca e0 0a   e7
1878 : 90 03 4c 28 7d 4c ec 60   51
1880 : a2 00 a0 00 e8 bd ff 0f   e5
1888 : f0 5c 85 78 86 77 a2 06   d1
1890 : bd 05 19 ca d0 0b 05 77   d8
1898 : c9 3a d0 05 20 0c 19 f0   fc
18a0 : 03 20 69 92 8a 10 e9 a2   76
18a8 : 07 b9 0a 10 c8 48 86 79   ec
18b0 : a2 04 dd a3 61 f0 34 ca   44
18b8 : d0 f8 a6 79 e0 08 90 07   7c
18c0 : d0 0a a9 2b 20 69 92 a9   50
18c8 : 22 20 69 92 68 20 69 92   f9
18d0 : a2 09 c6 78 d0 d3 e0 09   f9
18d8 : 90 05 a9 22 20 69 92 a9   84
18e0 : 8d 20 69 92 a6 77 e0 0a   e8
18e8 : d0 9a 60 a6 79 bd 9a 61   a5
18f0 : 20 69 92 ca e0 03 b0 f5   98
18f8 : 68 20 30 a8 a9 29 20 69   c9
1900 : 92 a2 08 d0 cd 2c 30 20   3f
1908 : 59 45 4b 46 a2 02 bd 18   01
1910 : 19 20 69 92 ca d0 f7 60   ba
1918 : 20 30 31 20 74 a0 a9 20   d4
1920 : 85 fb a9 00 85 fa a0 00   c0
1928 : b1 fa 49 ff 91 fa c8 d0   5e
1930 : f7 e6 fb a5 fb c9 40 d0   fe
1938 : ef 60 a9 00 85 0c 20 ae   58
1940 : 91 20 45 a8 ac 11 12 ad   3f
1948 : 10 12 38 e9 02 b0 01 88   67
1950 : aa a5 0c 4c 3c 91 a9 00   50
1958 : 85 0c a9 e6 20 c1 a3 20   09
1960 : 50 a7 a9 00 8d 1d 01 a0   f5
1968 : 05 a9 04 20 67 a6 a9 00   99
1970 : aa 20 87 92 4c 41 19 20   d2
1978 : 56 79 20 da 77 a9 a1 a0   7b
1980 : 19 4c 08 8a 20 56 79 20   ee
1988 : da 77 a9 a6 a0 19 4c 08   71
1990 : 8a 20 56 79 20 da 77 20   e6
1998 : ca 89 a9 ab a0 19 4c 08   1b
19a0 : 8a 7b 0e fa 35 0f 86 65   7b
19a8 : 2e e0 d2 7f 5e 5b d8 aa   64
19b0 : 20 f7 87 ca 86 fa 20 09   da
19b8 : 88 20 56 79 ca bd e3 19   71
19c0 : 85 fb a6 fa bd e3 19 05   b5
19c8 : fb 85 fb 78 ad e7 11 a0   33
19d0 : 00 8c e7 11 58 25 fb c5   5c
19d8 : fb d0 03 a0 01 2c a0 00   04
19e0 : 4c d4 84 01 02 04 08 10   58
19e8 : 20 40 80 20 56 79 20 f7   ee
19f0 : 87 ca bd e3 19 85 fb 78   67
19f8 : ad e8 11 a0 00 8c e8 11   9c
1a00 : 58 25 fb c5 fb d0 03 a0   36
1a08 : 01 2c a0 00 4c d4 84 20   05
1a10 : f4 87 ca 30 0e d0 16 20   80
1a18 : 0f 88 84 35 84 39 85 36   c8
1a20 : 85 3a 60 20 0f 88 8c 12   6a
1a28 : 12 8d 13 12 60 4c 28 7d   0c
1a30 : 20 12 88 84 fa 85 fb 20   18
1a38 : 0f 88 84 fc 85 fd e6 fc   2a
1a40 : d0 02 e6 fd 20 09 88 8a   0c
1a48 : a0 00 a2 fa 8e b9 02 ae   0c
1a50 : d5 03 20 77 ff e6 fa d0   63
1a58 : 02 e6 fb a6 fb e4 fd d0   21
1a60 : ee a6 fa e4 fc d0 e8 60   b7
1a68 : 20 f4 87 e0 10 b0 6d 86   4a
1a70 : fe 20 0f 88 84 fa 85 fb   81
1a78 : 20 0f 88 84 fc 85 fd e6   94
1a80 : fc d0 02 e6 fd 20 09 88   58
1a88 : e0 10 b0 50 86 3f 20 0f   a7
1a90 : 88 84 77 85 78 a2 77 8e   80
1a98 : b9 02 a6 fe a0 00 a9 fa   82
1aa0 : 20 74 ff a6 3f 20 77 ff   a2
1aa8 : e6 fa d0 02 e6 fb e6 77   58
1ab0 : d0 02 e6 78 a6 fb e4 fd   24
1ab8 : d0 e0 a6 fa e4 fc d0 da   30
1ac0 : 60 20 f7 87 20 56 79 ca   4f
1ac8 : 30 09 d0 10 a4 39 a5 3a   d2
1ad0 : 4c c9 84 ac 12 12 ad 13   46
1ad8 : 12 4c c9 84 4c 28 7d 20   4f
1ae0 : f7 87 20 56 79 ca 30 09   2e
1ae8 : d0 f2 a4 2f a5 30 4c c9   e1
1af0 : 84 a4 2d a5 2e 4c c9 84   3c
1af8 : 20 f4 87 ca 30 12 d0 dc   5e
1b00 : 20 0f 88 84 2f 84 31 84   3f
1b08 : 33 85 30 85 32 85 34 60   9b
1b10 : 20 0f 88 84 2d 85 2e 60   e3
1b18 : a2 59 a0 00 20 00 8c 20   03
1b20 : 5c 79 20 d7 77 20 56 79   01
1b28 : a9 59 a0 00 20 1e 8a 4c   5c
1b30 : fb 8c a2 59 a0 00 20 00   d0
1b38 : 8c 20 5c 79 20 d7 77 a2   fe
1b40 : 5e a0 00 20 00 8c 20 25   22
1b48 : 1b a9 5e a0 00 20 24 8a   8a
1b50 : a9 59 a0 00 4c 2e 88 20   67
1b58 : f3 1b a2 05 4c 3c 4d 20   3e
1b60 : 74 a0 20 45 a8 a9 04 aa   12
1b68 : 20 ba ff 20 c0 ff a2 04   88
1b70 : 20 c9 ff 24 90 30 e0 a9   5b
1b78 : 08 20 69 92 a9 0d 20 69   93
1b80 : 92 a9 00 a2 06 86 fa 8d   d7
1b88 : 34 11 8d 31 11 8d 32 11   37
1b90 : a5 fa 8d 33 11 a9 06 85   fe
1b98 : fb 20 e3 9c b1 8c a6 fb   42
1ba0 : 9d 00 01 ce 33 11 c6 fb   26
1ba8 : 10 ef ee 33 11 a0 07 a9   57
1bb0 : 01 a2 06 1e 00 01 2a ca   8e
1bb8 : 10 f9 48 a5 fa c9 c8 90   ce
1bc0 : 04 68 29 8f 24 68 20 69   0d
1bc8 : 92 88 10 e3 a9 08 18 6d   35
1bd0 : 31 11 8d 31 11 90 03 ee   93
1bd8 : 32 11 20 24 9d 90 b1 a9   98
1be0 : 0d 20 69 92 a5 fa 18 69   0f
1be8 : 07 85 fa c9 cc b0 04 a5   57
1bf0 : d3 f0 97 a9 0f 20 69 92   13
1bf8 : a9 0d 20 69 92 4c cc ff   1c
```

Listing 1. »xbasic128.obj« ist nur 2 KByte lang. Bitte mit dem MSE (Seite 158) im C64-Modus eingeben

```
ready.
1000 -;                     topass assemblerlisting
1010 -;                           hcopy128
1020 -;
1030 -;                   michael schmand, 03.04.1987
1040 -;
1050 -.base $0c00
1060 -;
1070 -.define xcord                    = $1131
1080 -.define ycord                    = $1133
1090 -.define bytadr                   = $8c
1100 -.define setbit                   = $9ce3
1110 -.define getadr                   = $9d24
1120 -.define bsout                    = $9269
1130 -.define buffer                   = $100
1140 -.define temp1                    = $fa
1150 -.define temp2                    = $fb
1160 -.define shift                    = $d3
1170 -;
1180 -hcerr               jsr lab7        ;drucker-kanal schliessen
1190 -                    ldx #$05
1200 -                    jmp $4d3c       ;ausgabe device not present error
1205 -;
1210 -hcopy               jsr $a074       ;test graphic-ram eingerichtet
1220 -                    jsr $a845       ;mapio
1230 -                    lda #$04        ;drucker-kanal oeffnen
1240 -                    tax
1250 -                    jsr $ffba
1260 -                    jsr $ffc0
1270 -                    ldx #$04
1280 -                    jsr $ffc9
1290 -                    bit $90         ;status testen
1300 -                    bmi hcerr       ;fehler --->
1310 -                    lda #$08        ;grafik-modus
1320 -                    jsr bsout
1330 -                    lda #$0d        ;return ausgeben
1340 -                    jsr bsout
1350 -;
1360 -                    lda #$0         ;x/y koordinaten
1370 -                    ldx #$06        ;auf null setzen
1380 -                    stx temp1
1390 -                    sta ycord+1
1400 -lab1                sta xcord
1410 -                    sta xcord+1
1420 -lab2                lda temp1
1430 -                    sta ycord
1440 -                    lda #$06
1450 -                    sta temp2
1460 -lab3                jsr setbit      ;byteadr im grafikspeicher holen
1470 -                    lda (bytadr),y  ;byte laden
1480 -                    ldx temp2
1490 -                    sta buffer,x    ;und zwischenspeichern
1500 -                    dec ycord       ;schon 7 byte?
1510 -                    dec temp2
1520 -                    bpl lab3        ;nein --->
1530 -                    inc ycord
1540 -                    ldy #$07        ;7 bytes aus grafik-speicher
1550 -lab4                lda #$01        ;in 8 bytes druckerformat mps 801
1560 -                    ldx #$06        ;umwandeln
1570 -lab5                asl buffer,x
1580 -                    rol
1590 -                    dex
1600 -                    bpl lab5
1610 -                    pha
1620 -                    --
1630 -                    cmp #$c8        ;in letzter druckzeile?
1640 -                    bcc ok          ;nein --->
1650 -                    pla             ;sonst bits mit ycord>199
1660 -                    and #$8f        ;ausblenden
1670 -                    .byte $24        ;ein byte im programm ueberspringen
1680 -ok                  pla
1690 -                    jsr bsout       ;byte ausgeben
1700 -                    dey
1710 -                    bpl lab4
1720 -                    lda #$08        ;xcord erhoehen
1730 -                    clc
1740 -                    adc xcord
1750 -                    sta xcord
1760 -                    bcc lab6
1770 -                    inc xcord+1
1780 -lab6                jsr getadr      ;noch im grafik-bereich?
1790 -                    bcc lab2        ;xcord < 319 --->
1800 -                    lda #$0d        ;sonst naechste zeile
1810 -                    jsr bsout
1820 -                    lda temp1       ;ycord erhoehen
1830 -                    clc
1840 -                    adc #$07
1850 -                    sta temp1
1860 -                    cmp #$cc        ;letzte druckzeile erreicht?
1870 -                    bcs lab7        ;ja --->
1880 -                    lda shift       ;abbruch durch shift/ctrl/comm?
1890 -                    beq lab1        ;nein --->
1895 -;
1900 -lab7                lda #$0f        ;drucker in normalmodus
1910 -                    jsr bsout
1920 -                    lda #$0d        ;return ausgeben
1930 -                    jsr bsout
1940 -                    jmp $ffcc       ;kanal schliessen
```

Listing 2. Das dokumentierte TOPASS-Quell-Listing der Hardcopy-Routine für den MPS 801. Sie kann an andere Drucker angepaßt oder ersetzt werden.

```
ready.
1000 -;                     topass assemblerlisting
1010 -;                           xMain.src
1020 -;                   michael schmand, 31.01.1987
1030 -;
1040 -.base $1300
1050 -;
1060 -;
1070 -.define ecrunch                  = $4321
1080 -.define ucrunch                  = $43b2
1090 -.define search                   = $43e2
1100 -.define elist                    = $516a
1110 -.define chrget                   = $0380
1120 -.define jmper                    = $0056
1130 -.define fnjump                   = $0057
1140 -;
1150 -init                ldx #$5         ;esc-vektoren setzen
1160 -iloop               lda vector,x
1170 -                    sta $030c,x
1180 -                    dex
1190 -                    bpl iloop
1200 -                    lda #<(execfn) ;esc-fn-vektor setzen
1210 -                    sta $02fc
1220 -                    lda #>(execfn)
1230 -                    sta $02fd
1240 -                    rts
1250 -;
1260 -vector              .word crunch     ;vektoren-tabelle
1270 -                    .word list
1280 -                    .word exec
1290 -;
1300 -crunch              pha             ;ac retten
1310 -                    lda #>(cmdlst) ;startadresse cmd-tabelle
1320 -                    ldy #<(cmdlst)
1330 -                    jsr search      ;wort suchen
1340 -                    bcc fncrunch    ;nicht gefunden -->
1350 -                    tax
1360 -                    pla             ;stack bereinigen
1370 -                    txa
1380 -                    and #$7f        ;token in ac
1390 -                    clc
1400 -                    adc #$27        ;kleinstes user-cmd-token
1410 -                    ldx #$00        ;cmd-flag
1420 -                    jmp ucrunch     ;token in basic-text einsetzen
1430 -;
1440 -fncrunch            lda #>(fnlst)  ;startadresse fn-tabelle
1450 -                    ldy #<(fnlst)
1460 -                    jsr search      ;wort suchen
1470 -                    bcs found       ;gefunden -->
1480 -;
1490 -notfound            pla             ;sonst: ac zurueck
1500 -                    sec             ;flag fuer normalen basic-befehl
1510 -                    jmp ecrunch     ;tokenisieren
1520 -;
1530 -found               tax
1540 -                    pla             ;stack bereinigen
1550 -                    txa
1560 -                    and #$7f        ;token in ac
1570 -                    clc
1580 -                    adc #$0b        ;kleinstes user-fn-token
1590 -                    ldx #$ff        ;fn-flag
1600 -                    jmp ucrunch     ;in basic-text einsetzen
1610 -;
1620 -list                ora #$80        ;msb fuer list setzen
1630 -                    inx
1640 -                    beq fnlist      ;fn-token? ja -->
1650 -;
1660 -                    sec             ;kleinstes user-cmd-token
1670 -                    sbc #$27        ;fuer tabellen-offset subtrahieren
1680 -                    tax
1690 -                    lda #>(cmdlst) ;startadresse cmd-tabelle
1700 -                    ldy #<(cmdlst)
1710 -                    jmp elist       ;--> list
1720 -;
1730 -fnlist              sec             ;kleinstes user-fn-token
1740 -                    sbc #$0b        ;fuer tabellen-offset subtrahieren
1750 -                    tax
1760 -                    lda #>(fnlst)  ;startadresse fn-tabelle
1770 -                    ldy #<(fnlst)
1780 -                    jmp elist       ;--> list
1790 -;
1800 -exec                and #$7f        ;msb ausblenden
1810 -                    sec             ;kleinstes user-cmd-token
1820 -                    sbc #$27        ;fuer tabellen-offset subtrahieren
1830 -                    asl             ;mal  2, da 2-byte adressen
1840 -                    tay
1850 -                    lda comadr+1,y ;routinen-adresse
1860 -                    pha             ;auf stack legen
1870 -                    lda comadr,y
1880 -                    pha
1890 -                    jmp chrget      ;--> rts ruft routine
1900 -;
1910 -execfn              sec             ;kleinstes user-fn-token
1920 -                    sbc #$0b        ;fuer tabellen-offset subtrahieren
1930 -                    asl             ;mal 2, da 2-byte adressen
1940 -                    tay
1950 -                    lda fnadr+1,y  ; routinen-adresse
1960 -                    sta fnjump+1    ;in sprungvektor eintragen
1970 -                    lda fnadr,y
1980 -                    sta fnjump
1990 -                    jsr jmper       ;routine aufrufen
2000 -                    clc             ;flag fuer user-fn ausgefuehrt
2010 -                    rts
2020 -;
2030 -comadr              .word fullscreen-1;adressliste befehle
2040 -;
2050 -;hier koennen weitere befehlsadressen-1 stehen
2060 -;
2070 -;
2080 -fnadr               .word ten        ;adressliste funktionen
2090 -;
2100 -;hier koennen weitere funktionsadressen stehen
2110 -;
2120 -;
2130 -cmdlst              .byte "fullscreeN";schluesselwoerter befehle
2140 -;                                      ,erstes token $fe $27
2150 -;
2160 -;hier koennen weitere schluesselwoerter stehen
2170 -;der letzte buchstabe muss immer geshiftet sein
2180 -;
2190 -                    .byte 0         ;tabellen-ende
2200 -;
2210 -fnlst               .byte "teN"     ;schluesselwoerter funktionen
2220 -;                                   ,erstes token $ce $0b
2230 -;
2240 -;hier koennen weitere schluesselwoerter stehen
2250 -;der letzte buchstabe muss immer geshiftet sein
2260 -;
2270 -                    .byte 0         ;tabellen-ende
2280 -;
2290 -;jetzt folgen die befehls-routinen (beispiele)
2300 -;
2310 -;
2320 -fullscreen          jmp $ca24       ;vollen Bildschirm herstellen
2330 -;
2340 -ten                 jsr $7956       ;test auf ")", immer!
2350 -                    jmp $8b17       ;fac mal zehn
2360 -;
2370 -;
2380 -;die folgenden werte koennen nicht als token benutzt werden:
2390 -;
2400 -;    $20, entspricht space (" ")
2410 -;    $22, entspricht anfuehrungszeichen
2420 -;    $3a, entspricht trennzeichen (":")
2430 -;
2440 -;an ihrer stelle ist in der adressliste .word 0 einzutragen,
2450 -;in der schluesselwortliste .byte $a0
```

Listing 3. Die verantwortliche Interpreter-Routine von xBasic 128 als TOPASS-Quelltext. Sie kann für eigene Basic-Erweiterungen verwendet werden.

Der freie Steckplatz des C128

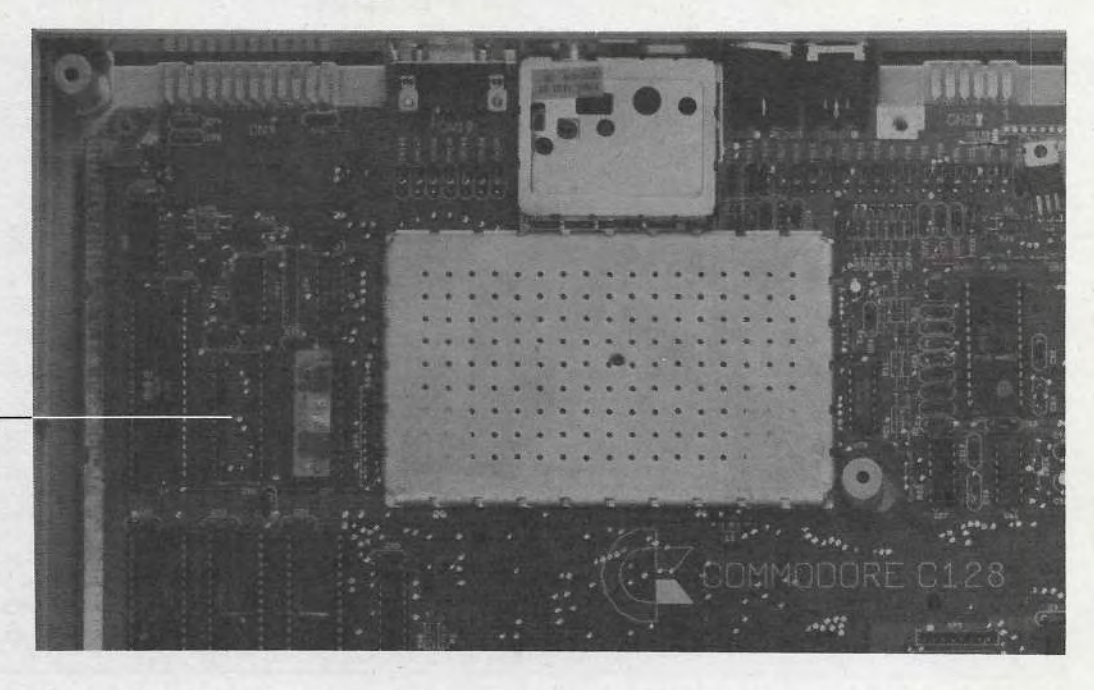


Bild 1. Die Platine des C128. Der freie Steckplatz neben dem ROM des C128-Betriebssystems kann beispielsweise für unseren »Double-Ass« verwendet werden.

Wenn Sie Ihren C128 schon einmal geöffnet haben, dann wird Ihnen sicherlich der leere IC-Sockel auf der Platine aufgefallen sein. Was man mit diesem alles anfangen kann, sagt Ihnen der folgende Artikel.

Neben einer Vielzahl an Bausteinen enthält der C128 auch einen Sockel, der beim Kauf des Geräts noch nicht belegt ist. Dieser freie Steckplatz eröffnet uns eine Vielzahl an Möglichkeiten. Schrauben Sie das Gehäuse Ihres C128 auf (Vorsicht! Garantieverlust), dann erkennen Sie den leeren Sockel auf der linken Platinenseite neben dem ROM des C128-Betriebssystems (Bild 1). Der Sockel trägt die Bezeichnung U 36 und ist für ein 16- oder 32-KByte-EPROM (27128 oder 27256) vorgesehen.

Vielfältige Möglichkeiten

Die Konfiguration auf der Hauptplatine des C 128D ist wie in Bild 1. Seit kurzem gibt es eine neue C 128D-Platine, auf der sich der Steckplatz an einer anderen Stelle befindet (Bild 2). Laut Commodore gibt es auch eine neue C128-Platine mit stark geänderter Hardware-Aufteilung. Da uns ein solches Gerät noch nicht zur Verfügung gestellt wurde, können wir Ihnen nur sagen, daß es der einzige freie Steckplatz im Gerät sein müßte. Vermutlich ist die Konfiguration mit der neuen C 128D-Platine identisch.

In dem freien Steckplatz läßt sich beispielsweise ein EPROM mit dem 8502/Z80-Assembler »Double-Ass« aus dieser Ausgabe einsetzen. Die hard- und softwaremäßigen Grundlagen, die dazu notwendig sind, wollen wir Ihnen hier vermitteln.

Wie Sie sicherlich wissen, kann man sowohl in den C64 als auch in den C128 Erweiterungsmodule in einen dafür vorgesehenen Steckplatz auf der Rückseite des Computers stecken. Diese Buchse nennt sich Erweiterungs- oder Expansion-Port.

Der C128 kann sowohl die Module eines C64 als auch seine »eigenen« erkennen und schaltet dabei in den entsprechenden Modus. Die Module des C64 werden dabei dadurch identifiziert, daß sie die EXROM- und die GAME-Leitung für ihre Zwecke verwenden. Sie schalten sich also hardwaremäßig ein.

Beim C128 gingen die Entwickler einen anderen Weg. Hier kommt die MMU (»Memory Management Unit« oder »Speicherverwaltungs-Einheit«) ins Spiel. Da der C128 mehrere Speicherbänke zu je 64 KByte verwalten kann, muß zwischen den einzelnen Bänken auch umgeschaltet werden, um auf jedes Byte des Speichers zugreifen zu können.

Diese Umschaltung regelt die MMU, wobei auf Wunsch eine bestimmte Konfiguration des Speichers eingestellt werden kann. Dazu muß man wissen, daß der Speicher des C128 in einen RAM-, einen unteren, einen mittleren und einen oberen Speicherbereich aufgeteilt ist. Dabei geht der untere Bereich von $4000 bis $7FFF, der mittlere von $8000 bis $BFFF und der obere von $C000 bis $FFFF. Der RAM-Bereich von $0000 bis $3FFF kann, wie der Name schon sagt, nur RAM enthalten. Die drei anderen Bereiche können jedoch wahlweise mit einer bestimmten Belegung versehen werden (Bild 3).

Wir wollen die Ausführungen an dieser Stelle zwar so knapp wie nur möglich halten, die wichtigsten Hinweise zur Einstellung der Speicherkonfiguration sollen jedoch an dieser Stelle gegeben werden. Für detailliertere Informationen kann das Handbuch zum C128 zu Rate gezogen werden.

Für die Einstellung der Speicherkonfiguration existiert in der MMU ein wichtiges Register: das **Konfigurationsregister**, dessen Belegung in Tabelle 1 dargestellt wird. Es ist die Adresse $D500, gespiegelt bei SFF00.

Speicherkonfiguration selbst bestimmen

Die Bit-Belegung im einzelnen:

Bit 7/6: Bit 7 und 6 geben die Nummer der aktuellen RAM-Bank an, die für die gesamten 64 KByte gilt, die gerade bearbeitet werden. Da in der Grundversion des

C128 nur die Bänke 0 und 1 existieren, können wir Bit 7 vernachlässigen. Bit 6 gibt also die Banknummer an.

Der RAM-Bereich von $0000 bis $3FFF ist, wie schon gesagt, immer mit RAM belegt. Dabei ist die Banknummer in Bit 6 ausschlaggebend.

Bit 5/4: Bei den Bits 5 und 4 existieren vier verschiedene Möglichkeiten. Sind beide Bits gelöscht, so ist im oberen Bereich der aktuellen Bank von $C000 bis $FFFF das Betriebssystem-ROM (Kernel) des C128 eingeblendet. Es steckt auf der Platine an Position U 35 und enthält 16 KByte Speicher. Sind die Bits 5 und 4 beide gesetzt, so adressiert die MMU im betreffenden Speicherbereich 16-KByte-RAM der entsprechenden Speicherbank.

So, und jetzt wird es interessant: Sie haben nämlich jetzt noch zwei Bit-Kombinationen übrig, die die Bits 4 und 5 annehmen können. Ist Bit 5 gesetzt und Bit 4 gelöscht, so wird der Expansion-Port des C128 adressiert. Steckt hierin nun ein Modul, so wird es in den Speicherbereich $C000 bis $FFFF eingeblendet und behandelt, als wäre es ein ganz »normales« Kernel.

Ist der Zustand der Bits 5 und 4 hingegen umgekehrt, also Bit 5 gelöscht und Bit 4 gesetzt, so wird der interne Zusatzsteckplatz U 36, der uns im folgenden noch genauer interessieren soll, adressiert. Hier kann nun zum Beispiel ein völlig neues Betriebssystem oder eine Anwendungs-Software integriert sein.

Wie Sie sehen, haben im C128 sowohl ROM und RAM als auch Module und ROMs im Zusatzsteckplatz die gleiche Priorität. Es wird lediglich im Konfigurationsregister das eingestellt, was der Anwender gerne als Kombination haben möchte.

Bit 3/2: Die Bits 3 und 2 haben genau die gleiche Funktion wie die Bits 5 und 4. Hier wird lediglich der Adreßbereich von $8000 bis $BFFF geschaltet, so daß Sie weitere Kombinationsmöglichkeiten haben. Es steht Ihnen also zum Beispiel frei, von $8000 bis $BFFF ein Modul einzuschalten, während der Bereich $C000 bis $FFFF RAM enthält. Der Speicherbereich von $8000 bis $BFFF wird übrigens normalerweise durch ein ROM repräsentiert, das einen Teil des Basic-Interpreters und den TEDMON des C128 enthält. Es ist auf der Platine im Sockel U 34 untergebracht.

Bit 1: Bit 1 ist für den Adreßbereich $4000 bis $7FFF zuständig. Hier können Sie kein Modul oder externes ROM mehr einblenden, sondern haben nur noch die Wahl zwischen Basic-ROM oder RAM. Bei dem Basic-ROM handelt es sich um den ersten Teil des Interpreters, der in einem 16-KByte-ROM in dem Steckplatz U 33 enthalten ist.

Bit 0: Bit 0 des Konfigurationsregisters hat eine Sonderfunktion, die den Adreßbereich $C000 bis $FFFF betrifft. Im Bereich von $D000 bis $DFFF kann nicht nur RAM oder ROM eingeblendet werden; hier liegen vielmehr auch noch die I/O-Bausteine des C128.

Soll also ein Zugriff auf VIC, SID, CIA oder andere Peripherie erfolgen, so muß von $D000 bis $DFFF der Ein-/Ausgabebereich des Computers eingeblendet werden. Ist Bit 0 gelöscht, so ist der I/O-Bereich eingeblendet, wobei sich gleichzeitig der Speicherbereich infolge der ausgefallenen 4 KByte auf 12 KByte vermindert ($C000 bis $CFFF und $E000 bis $FFFF). Ist das Bit gleich eins, so kann auf den zusätzlichen Speicherbereich zugegriffen werden.

Schön und gut, werden Sie jetzt sagen. Man kann also auf das ROM im Zusatzsteckplatz (das wir übrigens ab jetzt als Funktions-ROM bezeichnen) zugreifen. Toll, aber was bringt mir das?

Auf das Funktions-ROM können Sie nicht nur zugreifen, es wird vielmehr vom Betriebssystem des C128 unterstützt. Das »Wie« soll uns dabei nun interessieren.

Wie bereits erwähnt, werden Module beim C128 nicht hardwaremäßig initialisiert und gestartet. Der C128 regelt das Einschalten eines Moduls sehr viel eleganter. Verfolgen wir einmal den Verlauf einer solchen Initialisierung.

Bit	Erklärung
7	höherwertiges Bit der Nummer der Speicherbank
6	niederwertiges Bit der Nummer der Speicherbank
5/4	Adreßbereich $C000 bis $FFFF:
	0/0 Kernel eingeblendet
	0/1 Funktions-ROM eingeblendet
	1/0 Steckmodul im Expansion-Port eingeblendet
	1/1 RAM eingeblendet
3/2	Adreßbereich $8000 bis $BFFF:
	0/0 Basic-ROM 2 eingeblendet
	0/1 Funktions-ROM eingeblendet
	1/0 Steckmodul im Expansion-Port eingeblendet
	1/1 RAM eingeblendet
1	Adreßbereich $4000 bis $7FFF:
	0 Basic-ROM 1 eingeblendet
	1 RAM eingeblendet
0	Adreßbereich $D000 bis $DFFF (I/O):
	0 I/O-Bausteine eingeblendet
	1 Konfiguration von Bit 4 und 5 eingeblendet

Tabelle 1. Die Belegung des Konfigurationsregisters in der MMU des C128. Die Grundadresse ist $D500 beziehungsweise gespiegelt bei Adresse $FF00.

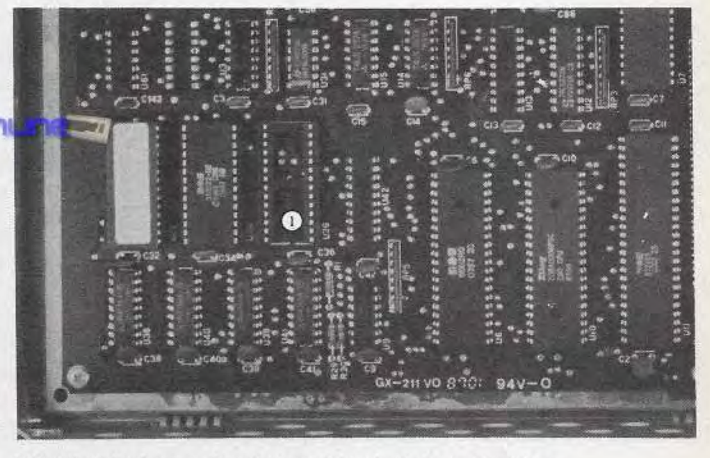


Bild 2. Die Lage des freien Steckplatzes U36 im neuen C128D ist mit einer »1« gekennzeichnet

CBM-Kennung abfragen: Wird ein Reset ausgeführt, so blendet das Betriebssystem des Computers mit Hilfe des Konfigurationsregisters nacheinander vier verschiedene Erweiterungseinheiten in die aktuelle Bank ein: Funktions-ROM von $8000 bis $BFFF, Funktions-ROM von $C000 bis $FFFF, Expansion-Port von $8000 bis $BFFF und Expansion-Port von $C000 bis $FFFF. Es wird also jetzt beispielsweise das Funktions-ROM bei $8000 eingeblendet. Dann liest das Betriebssystem die Speicherstellen $8009, $800A und $800B aus und schaut nach, ob dort die Kennung »CBM« enthalten ist. Ist das nicht der Fall, so wird der Adreßbereich $C009 bis $C00B untersucht. Findet der C128 auch hier nichts, prüft er noch die gleichen Adreßräume für den Expansion-Port und führt bei einem Mißerfolg die »normale« Reset-Routine aus, um schließlich das Basic 7.0 zu starten.

Wir haben also auch beim C128 die Abfrage auf eine CBM-Kennnung, die wir schon vom C64 her kennen.

Wie beim C64, so erfolgt auch beim C128 ein Start des entsprechenden Moduls, wenn die Bedingungen dafür erfüllt sind. Hier sind jedoch noch ein paar Einzelheiten zu beachten.

Beim C64 wird im Fall eines Modulstarts indirekt an die Adresse gesprungen, die in $8000 und $8001 gespeichert ist. Der C128 springt hingegen direkt in das Modul oder das Funktions-ROM. Wurde also die Kennung festgestellt, so erfolgt der Sprung an die Grundadresse, bei der das Modul oder ROM gefunden wurde. Diese Grundadresse ist je nach gefundener Kennung entweder $8000 oder $C000.

Direkter Sprung ins Modul

Theoretisch kann mit dem Befehl RTS in das Betriebssystem des C128 zurückgesprungen werden, da ein JSR in das Funktions-ROM stattgefunden hat. Es ist hierbei jedoch zu beachten, daß das Funktions-ROM anders adressiert wird als das Kernel, so daß das Kernel unter Umständen ausgeblendet wird, um das Funktions-ROM einzuschalten. Der C128 muß also immer mit sogenannten »long-jumps« arbeiten, was einem Umweg über das RAM entspricht.

Das Byte vor der CBM-Kennung: Selektierungsart

Neben der CBM-Kennung im Funktions-ROM ist aber auch noch ein anderer Faktor für einen Sprung in das ROM ausschlaggebend. Es handelt sich hierbei um ein Byte, das immer genau vor der CBM-Kennung zu finden ist, also bei $8008 beziehungsweise $C008.

- Steht dieses Byte auf $00, so wird das eingesteckte Funktions-ROM oder Modul nicht als Autostart-Modul erkannt, und es erfolgt kein Sprung in das eingesteckte ROM oder EPROM.
- Steht das Byte hingegen auf $01, so wird das Funktions-ROM oder das eingesteckte Modul direkt in der Reset-Routine angesprungen, und zwar bevor der gesamte Computer in einen definierten Zustand versetzt worden ist. Hier hat man also sehr früh die Möglichkeit, in den Betrieb des C128 einzugreifen.
- Steht das Byte in $8008 ($C008) hingegen auf $02 oder einem beliebigen anderen Wert bis $FF, so geschieht die dritte Möglichkeit der Ansteuerung des Funktions-ROM.

Das Interessante an dem ROM ist nämlich, daß bei Vorhandensein der Kennung auch an späterer Stelle der Reset-Routine noch ein Einsprung erfolgen kann.

Wenn Sie Ihren C128 einschalten und das Diskettenlaufwerk schon vorher eingeschaltet war, so erfolgt die Meldung des Basic 7.0 auf dem Bildschirm. Bevor jedoch das »READY.« mit dem Cursor erscheint, erfolgt ein Diskettenzugriff, der eine bootfähige Diskette erkennen und gegebenenfalls ein Programm laden und automatisch starten soll.

Betrachtet man sich die Routinen des Betriebssystems jedoch genauer, so kann man erkennen, daß vor dem Diskettenzugriff noch einmal die vier möglichen ROM-Variationen abgefragt werden, so wie es direkt nach dem Einschalten des Computers passiert.

Hier kann der Programmierer nun ansetzen, wenn er das Steuerbyte in $8008 ($C008) mit einem Wert zwischen $02 und $FF versieht. Wird dieser Wert erkannt, so startet der C128 das Zusatz-ROM vor dem Zugriff auf die Diskette, aber erst nachdem bereits das gesamte C128-Computersystem initialisiert wurde.

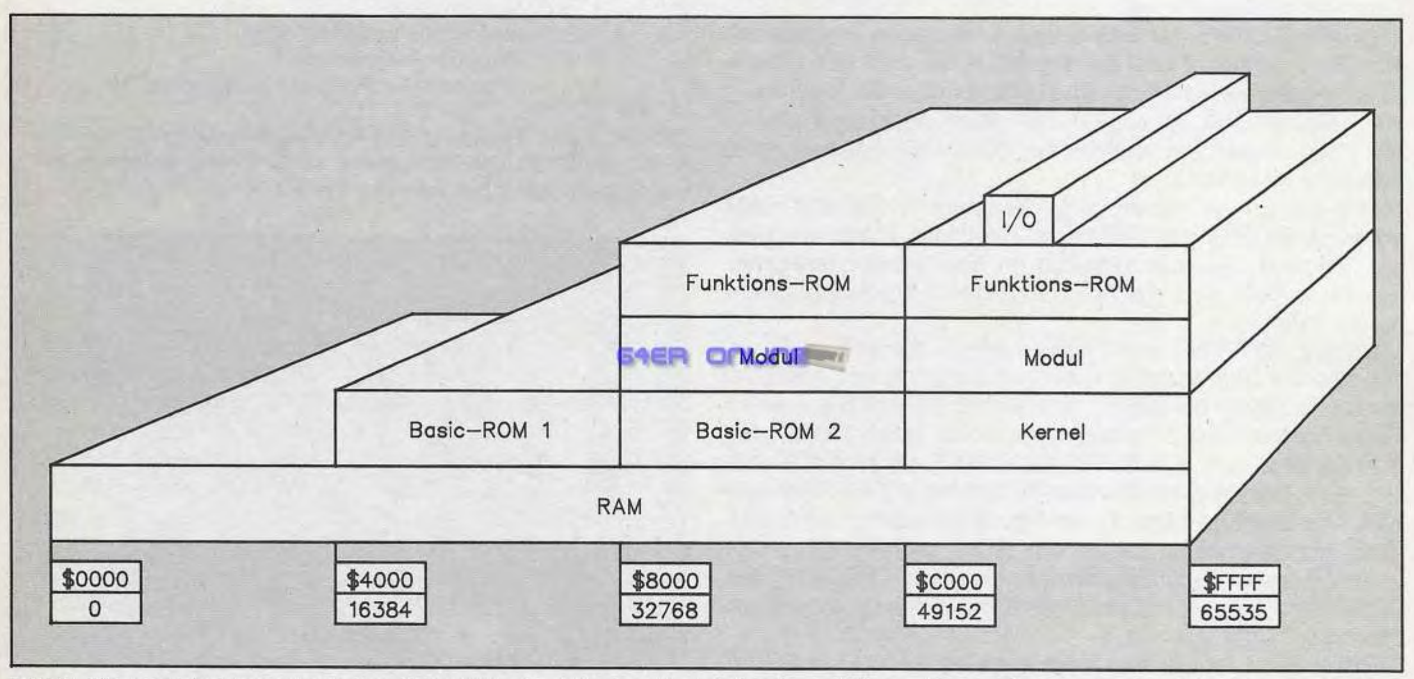


Bild 3. Speicherkonfiguration des C 128. Der Speicher teilt sich in einen RAM-, einen unteren, einen mittleren und einen oberen Bereich auf. Zusätzlich liegt bei $D000 der I/O-(Input/Output)-Bereich.

Sie haben als Anwender also drei Möglichkeiten: Entweder Sie programmieren sich ein eigenes Betriebssystem, das eine komplett eigene Reset-Routine und Initialisierung des Computers enthält. Dann ist es sinnvoll, wenn Sie direkt am Anfang der Reset-Routine in das eigene System springen (Byte also auf $01 setzen). Wollen Sie jedoch kein Betriebssystem entwickeln, sondern nur eigene Maschinenprogramme in ein Modul einbinden, so ist es besser, wenn Sie Ihr Programm erst dann starten lassen, wenn sich das gesamte Computersystem bereits in einem definierten Zustand befindet (Byte-Wert $02 bis $FF). Es ist jedoch auch möglich, daß Sie Software besitzen, bei der es angebracht ist, einen Teil in ein Modul und einen anderen Teil auf Diskette zu haben. Dann wäre es am günstigsten, wenn Sie den Wert $00 in das Steuerbyte schreiben und das Modul mit Ihrem eigenen Programm, nachdem dies von Diskette geladen wurde, aktiv einschalten und anspringen.

Einblenden in verschiedenen Varianten

Ein kleiner Fehler im Betriebssystem des C128 wäre vielleicht noch erwähnenswert: Wenn Sie die Kennung $01 im

Steuerbyte stehen haben und aus der frühen Initialisierung mit RTS wieder in die Reset-Routine des C128 zurückspringen, so erfolgt der Aufruf des Moduls noch einmal beim Basic-Start. Sie bekommen also zwei Autostarts Ihres Moduls, da der zweite Start nur noch auf das Steuerbyte ungleich Null abfragt und nicht verlangt, daß ein Wert größer gleich $02 vorhanden ist. Dieser Fehler wird uns aber in der Regel nicht stören, da ein Rücksprung aus dem Modul in das Betriebssystem in den wenigsten Anwendungsfällen sinnvoll ist.

Freie Bereichswahl

Wenn Sie ein EPROM mit 16 KByte Speicherkapazität in den Sockel U 36 einstecken, haben Sie natürlich im Bereich ab $8000 und ab $C000 den gleichen Inhalt bei der Initialisierung. Sie können sich also hier auswählen, ob Sie den $8000 bis $BFFF- oder den $C000 bis $FFFF-Bereich als Modulbereich anwählen. In den meisten Fällen wird $8000 bis $BFFF der günstigere Bereich sein, da man dann das Betriebssystem des Computers ab $C000 einblenden und aus dem Modul heraus sehr einfach anspringen kann.

Haben Sie hingegen ein 32-KByte-EPROM im Sockel stecken, so muß natürlich der gesamte Bereich von $8000 bis $FFFF auf Funktions-ROM geschaltet werden, um einen Zugriff zu ermöglichen. Ob man dann die Startadresse bei $8000 oder bei $C000 liegen hat, ist nur eine Frage des eigenen Geschmacks und hat für den Betrieb des EPROM keine Bedeutung.

Eine mächtige Angelegenheit also, das Funktions-ROM. Sie können von der eigenen Programmiersprache bis hin zu eingebauten DFÜ-Systemen oder einem Assembler wie Double-Ass alles realisieren, was Sie wollen. Und das alles steht auf Wunsch, direkt nach dem Einschalten des Computers, ohne einen Verlust des Original-Betriebssystems zur Verfügung. (Karsten Schramm/kn)

Z80 – Der CP/M-Steuermann

Lernen Sie die zweite Seele des C128, den Z80-Prozessor, näher kennen. Mit dem »Double-Ass« aus dieser Ausgabe können Sie auch diesen Prozessor leicht programmieren. Hier finden Sie wichtige Grundlagen dafür.

Wer sich heute ernsthafter mit CP/M beschäftigen oder den zweiten Prozessor im C128 mit Hilfe von »Double-Ass« (Seite 66) individuell nutzen will, kommt nicht darum herum, den Z80 Mikroprozessor etwas näher zu betrachten.

Das Betriebssystem CP/M (»C«ontrol-»P«rogramm for »M«icrocomputers), das von der Firma Digital Research entwickelt wurde, ist zwar ursprünglich für den 8080-Mikroprozessor der Firma Intel geschrieben worden, doch heutzutage wird in CP/M-Computern fast ausschließlich der Z80-Mikroprozessor (CPU) verwendet. So wird auch im Commodore 128 PC eine Z80 A CPU (A = 4-MHz-Version) für den CP/M-Modus eingesetzt.

Kompatibilität Z80 – 8080

Die Kompatibilität der 80er-Prozessoren von Intel und Zilog stiftet bei CP/M-Benutzern immer wieder Verwirrung. Deshalb wird an dieser Stelle hierauf eingegangen.

Die 8080-CPU der Firma Intel kann als der Stammvater der sogenannten 80er-Familie bezeichnet werden. Diese CPU ist seit 1974 lieferbar. Die Z80-CPU wurde von der Firma Zilog entwickelt und wird seit 1977 vertrieben. Dieser Mikroprozessor ist als konsequente Weiterentwicklung des 8080-Mikroprozessors zu sehen. Der Z80 ist aufwärtskompatibel zum 8080-Mikroprozessor, das heißt, er »versteht« alle Befehle des 8080 und besitzt darüber hinaus einen stark erweiterten Befehlssatz. Dieser erweiterte Befehlssatz darf aber für CP/M-Anwenderprogramme nicht benutzt werden, da Programme dann nicht mehr auf allen CP/M-Rechnern laufen würden (keine Kompatibilität mehr). Die Nutzung des erweiterten Befehlssatzes ist also auf Einzelanwendungen beschränkt. Einzelne Softwarehäuser weichen jedoch heute von dieser Regelung ab und bieten spezielle CP/M-Programme für CP/M- Rechner mit Z80-CPU an. Diese Softwarehäuser müssen sich jedoch den Vorwurf der »Nichtkompatibilität« gefallen lassen.

Wer nun glaubt, er könne einen defekten Z80 durch einen 8080 austauschen, der irrt gewaltig. Die Kompatibilität dieser beiden Prozessoren beschränkt sich auf den Befehlssatz des 8080, das heißt, der Z80 versteht die Opcodes des 8080. Hardwaremäßig besteht keine Kompatibilität. Der 8080 (und die 8080-Nachfolger der Firma Intel) benutzt einen gemultiplexten Datenbus, das heißt auf den Adreßleitungen A0-A7 werden abwechselnd Adressen und Daten »transportiert«, während beim Z80 alle Anschlüsse einzeln herausgeführt werden.

Die Verwirrung bei CP/M-Einsteigern ist häufig vollends, wenn zur Erstellung von Maschinenprogrammen Assembler benutzt werden sollen.

Der Besitzer eines C128 weiß, daß sein Computer mit einer Z80-CPU ausgerüstet ist und das Betriebssystem CP/M Plus mitgeliefert wurde. Um den Z80 näher kennenzulernen und in Maschinensprache zu arbeiten, besorgt er sich ein Buch über den Z80 und die zu CP/M Plus gehörenden Utilities SID, MAC und HEXCOM. SID ist ein Maschinensprachemonitor mit kleinem Assembler und Disassembler. MAC ist ein Makroassembler, der ein Maschinenprogramm in Assemblersprache (mnemonische Form) in den Maschinencode übersetzt. HEXCOM erzeugt aus den von MAC übersetzten Dateien Programme, die direkt vom Betriebssystem aus aufgerufen werden können (.com-Dateien). Wenn man sich jetzt mit SID einen Teil des Z80-ROMs in seinem C128 disassemblieren läßt, stellt man fest, daß die aufgelisteten Maschinensprachebefehle in mnemonischer Form nicht mit denen eines Z80-Buches übereinstimmen. Das liegt daran, daß die Opcodes des Z80 und des 8080 zwar übereinstimmen (mit den weiter oben beschriebenen Einschränkungen), jedoch für beide Prozessoren andere Mnemonics verwendet werden. Die zum CP/M-System mitgelieferten Disassembler erzeugen jedoch Mnemonics im Intel-Format (8080-Mnemonics).

Wer also den Z80 kennenlernen möchte, muß entweder mit Opcodes arbeiten (zum Beispiel mit dem SID-Monitor) oder einen Z80-Assembler wie Double-Ass benutzen. Da der Double-Ass brandneu ist, wurden die weiter unten

beschriebenen Z80-Assemblerbeispiele jedoch mit dem Z80-Makroassembler erstellt und getestet, der mit dem Small-C-Entwicklungssystem vom Markt & Technik Verlag mitgeliefert wird.

Der CP/M-Macroassembler MAC kann zwar auch Z80-Mnemonics übersetzen, jedoch muß eine Mnemonic-Form benutzt werden, die sehr stark von der üblichen Zilog-Version abweicht, so daß dieser Weg nur als Notlösung akzeptiert werden kann. Welche Form die zu verwendenden Mnemonics haben müssen, können Sie erfahren, wenn Sie die entsprechende Diskette einlegen (CP/M Utilities-Disk 2) und dann »TYPE Z80.LIB« (<RETURN>) eingeben. Oder Sie schauen sich die Mnemonics einfach beim Double-Ass in dieser Ausgabe an.

Der Z80-Macroassembler des Small-C-Entwicklungssystems sowie der Double-Ass benutzen die übliche Zilog-Version für Z80-Mnemonics.

Diese Vorbemerkungen sind notwendig, damit die ersten Versuche, praktisch mit dem Z80 zu arbeiten, nicht in Verwirrung und Frustration enden.

Aufbau des Z80

Der Z80-Mikroprozessor besitzt einen 8 Bit breiten Datenbus (deshalb 8-Bit-Prozessor), einen 16 Bit breiten Adreßbus und einen Steuerbus. Der Systemtakt muß extern von einem separaten Taktgenerator erzeugt werden. Der prinzipielle interne Aufbau des Z80-Prozessors sieht wie folgt aus:

Im Inneren des Prozessors befinden sich:
- eine Arithmetik-Logik-Einheit (ALU)
- ein Steuerwerk, das die internen Abläufe steuert sowie den Steuerbus verwaltet
- vier reine Adreßregister (1. der Befehlszähler PC, 2. der Stapelzeiger SP, 3. das Indexregister IX und 4. das Indexregister IY)
- ein Refresh-Register (R-Register), das dazu dient, dynamische Speicher aufzufrischen.
- zwei Akkumulatoren (A und A'), von denen jedoch nur einer zur Zeit benutzt werden kann. Auf den zweiten Akku kann jedoch umgeschaltet werden.
- zwei Registersätze mit »normalen« Registern (B-,C-,D-,E-,H- und L-Register beziehungsweise B'-, C'-,D'-,E'-,H'- und L'-Register). Diese jeweils 8 Bit breiten Register können zu jeweils drei 16 Bit breiten Registern zusammengezogen werden (BC-,DE- und HL-Register).

  Auch hier ist eine Registersatzumschaltung möglich wie bei den Akkumulatoren. Alle Register finden Sie in Bild 1 nochmals in einer Übersicht.
- interne Daten-, Adreß- und Steuerbusse. Diese internen Busse sind durch Puffer von den äußeren Bussen getrennt.

Es existieren noch weitere interne Register, die jedoch für den Programmierer nicht direkt erreichbar sind.

Unterschiede zwischen dem Z80 und dem 6502

Vergleicht man den Z80 mit dem 6502 (bzw. mit der Befehls-kompatiblen 2-MHz-Version 8502 im C128), so fällt die größere Anzahl von Registern beim Z80 auf. Der Z80 benötigt auch mehr Register, denn er gehört zu den sogenannten registerorientierten Prozessoren (wie die gesamte 80er-Familie), während der 6502 (8502) den Speicherorientierten Prozessoren zugeordnet wird. Wenn wir ab jetzt vom 6502-Prozessor sprechen, treffen die damit verbundenen

Hauptregister		Sekundäre Register	
Akkumulator A	Flags F	Akkumulator A'	Flags F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'

Befehlszähler	PC
Stapelzeiger	SP
Index-Register	IX
Index-Register	IY

Unterbrechungs-vektor 1
Speicher-Auf-frisch-Register A

Bild 1. Übersicht der verschiedenen Z80-Register

Aussagen gleichzeitig auf den, wie bereits gesagt, Befehls-kompatiblen 8502 im C128 zu.

Viele Arbeiten, die die speicherorientierten Prozessoren im Arbeitsspeicher (RAM) durchführen, müssen bei den registerorientierten Prozessoren innerhalb des Prozessors in einem ihrer vielen Register durchgeführt werden. Die 6502 CPU behandelt Ein-/Ausgabebausteine (I/O) wie normale Speicherzellen. Das bedeutet, daß die meisten CPU-Befehle auch auf I/O-Bausteine anwendbar sind.

Die Z80 CPU benutzt für Ein-/Ausgabe über I/O-Bausteine besondere Befehle (IN und OUT). Die I/O-Bausteine befinden sich nicht im »normalen« Adreßbereich, sondern werden über eine besondere Input/Output-Request-Leitung (IORQ) aktiviert und über die untere Hälfte des Adreßbusses (A0-A7) adressiert. Nach einem Hardware-Reset (zum Beispiel nach dem Einschalten) beginnt die 6502 CPU mit der Arbeit an der Speicherstelle FFFC, während die Z80 CPU bei der Adresse 0000 beginnt.

Weiterhin unterscheiden sich die beiden Prozessoren in den üblicherweise verwendeten Taktfrequenzen. Diese werden häufig fälschlicherweise zum Vergleich der Arbeitsgeschwindigkeit von unterschiedlichen Prozessoren herangezogen. Ein Vergleich ist direkt nur innerhalb gleicher Prozessorfamilien möglich.

Die Prozessoren der 80er-Familie arbeiten normalerweise mit wesentlich höheren Taktfrequenzen als die Prozessoren der 65er-Familie. Wir wollen den 8502 hier der 65er-Familie zuordnen. Die Schlußfolgerung, daß die 80er-Prozessoren damit grundsätzlich schneller sind als die 65er-Prozessoren, wäre aber falsch. Ein Mikroprozessor benötigt zur Abarbeitung eines Befehls in der Regel mehrere Taktzyklen. Vergleicht man nun Befehle des Z80 und des 6502, die die gleiche Wirkung haben, so stellt man fest, daß der Z80 in der Regel mehr Taktzyklen für den wirkungsgleichen Befehl benötigt als der 6502-Prozessor. Er muß, um auf die gleiche »Arbeitsgeschwindigkeit« zu kommen, mit einer höheren Taktfrequenz arbeiten.

Der Z80-Prozessor ist trotzdem verhältnismäßig schnell, weil er eine Methode benutzt, die als Pipelining bezeichnet wird. Vereinfacht ausgedrückt, bewirkt dieses sogenannte Pipelining, daß der Prozessor, während er einen Befehl bearbeitet, schon den nächsten Befehl lädt.

Die Befehle des Z80 lassen sich wie beim 6502 in bestimmte Befehlsgruppen einteilen.

1. Transportbefehle (auch Transferbefehle genannt)
2. Arithmetische Befehle (Addition)

3. Logische Operationen (UND-Verknüpfung)
4. Registeranweisungen (Übertragsbit löschen)
5. Sprungbefehle (Setze das Programm an der angegebenen Adresse fort)
6. Programmunterbrechungen (Unterbrechung wird zugelassen)
7. Unterprogrammbehandlung (Unterprogrammaufrufe)
8. Sonstige Befehle (NOP, Leerbefehl no operation)

Wie bekannt, lassen sich mit 8 Bit maximal 256 Werte darstellen (0-255), das heißt, der Befehlssatz eines 8-Bit-Mikroprozessors kann normalerweise maximal 256 Befehle betragen. Der 6502 besitzt 151 erlaubte (legale) Befehle (Opcodes) und in der C-MOS-Version 65C02 178 Befehle. Die restlichen Opcodes (bis zu einer Gesamtsumme von 256) sind die sogenannten »illegalen Opcodes«, für deren Funktion die Hersteller keine Garantie übernehmen.

Auch der 8080 bleibt mit der Anzahl seiner Befehle (242) im Rahmen der mit 8 Bit darstellbaren Werte.

Der Z80 hingegen besitzt mehr als 700 bekannte Befehle. Theoretisch sind es sogar mehr als 1000. Wie ist das bei einem 8-Bit-Mikroprozessor, der auch noch aufwärtskompatibel zum 8080/8085 ist, überhaupt möglich? Die Lösung ist einfacher, als man glaubt. Beim Z80 werden einige Opcodes, die beim 8080/8085 nicht benutzt werden, zur Umschaltung des gesamten Befehlssatzes verwendet. Und zwar sind es die Opcodes DD, FD, ED und CB (hexadezimale Schreibweise). Findet der Z80 einen dieser Opcodes in einem Programm vor, schaltet er seinen Befehlssatz um. Das diesen Opcodes folgende Byte wird dann als neuer Opcode interpretiert. Diese neuen Befehle bestehen aus mindestens 2 Byte (dem »Umschaltopcode«, gefolgt von einem Befehl). Diese genial einfache Lösung zur Erweiterung des Befehlssatzes für einen 8-Bit-Prozessor scheint den Entwicklern des Z80 so gut gefallen zu haben, daß sie diese Methode noch weiter ausgebaut haben und bei den »Umschalt-Opcodes« DD und FD noch weitere Umschalt-Opcodes vorgesehen haben. Folgt dem DD oder dem FD ein CB, wird jeweils wieder in einen neuen Befehlssatz umgeschaltet. Diese Befehle bestehen dann mindestens aus 3 Byte (3-Byte-Befehle).

Die beim Vorgänger des Z80 (dem 8080) nicht benutzten Opcodes DD, FD, ED, CB, DDCB und FDCB schalten also beim Z80 den gesamten Befehlssatz um. So ist es also möglich, mit einem 8-Bit-Prozessor einen Befehlssatz mit mehr als 256 Befehlen zu erzeugen. Nach diesem Prinzip funktioniert auch das sogenannte »Bank-switching«, die Speicherbankumschaltung bei 8-Bit-Computern mit mehr als 64-KByte-Speicher (wie auch beim C128).

Die Firma Zilog hat für den Z80 eine eigene Form von Mnemonics (Gedächtnishilfe für Opcodes) entwickelt, die stark von den Intel-Mnemonics abweichen. Wer sich schon mit der 6502-Maschinensprache beschäftigt hat und die 6502-Mnemonics kennt, kann sich aber freuen. Viele der Z80-Mnemonics ähneln stark den 6502-Mnemonics.

6502-Befehl:

```
LDA $1000
```

Wirkung: Lade den Akku mit dem Inhalt der Speicherstelle an Adresse 1000 hexadezimal.

Z80-Befehl:

```
LD A,(1000h)
```

Wirkung: gleiche Wirkung wie beim 6502. Die Klammer um die Adresse steht bei den meisten Z 80-Assemblern für: Inhalt der Speicherstelle mit der Adresse in der Klammer in den Akku übertragen. Weiterhin erwarten die meisten Z80-Assembler eine Angabe, in welchem Zahlensystem die Adressenangabe erfolgt. In diesem Beispiel besagt das »h« hinter der Adresse, daß es sich um eine hexadezimale Adressenangabe handelt. Unser Assembler »Double-Ass« verlangt jedoch statt einem »h« ein »$« am Anfang der Zahl, um sie als Hex-Zahl kenntlich zu machen.

Doch nun zu den Standardbefehlen der einzelnen Befehlsgruppen.

Z80-Mnemonics

Es hat wenig Sinn, im Rahmen dieses Artikels sämtliche Z80-Befehle (über 700) aufzulisten und in ihrer Wirkung zu beschreiben. Selbst im Rahmen eines Z80-Lehrbuches wäre diese Methode nicht nur langweilig, sondern auch unsinnig, da viele Befehle im Prinzip die gleiche Wirkung haben, sich jedoch nur auf andere Register beziehen. Deshalb werden hier die Standardbefehle der einzelnen Befehlsgruppen besprochen. Dabei wird das Prinzip dieser Befehle deutlich, das allen anderen der gleichen Kategorie zugrunde liegt, um den Anwender (beziehungsweise den Lernenden) in die Lage zu versetzen, alle Befehle dieser Art bei Bedarf einzusetzen. Um größere Programme zu erstellen, sollte man sich dann einen kompletten Befehlssatz des Z80 besorgen und die geeignetsten Befehle einsetzen.

Die für den Einstieg in die Z80-Programmierung wichtigen Befehlsgruppen werden im folgenden behandelt.









Mit Hilfe von Transferbefehlen können
- Daten vom Prozessor (aus einem seiner Register) zum Speicher (RAM) und umgekehrt vom Speicher (RAM oder ROM) zum Prozessor
- Daten vom Prozessor zu einem Ein-/Ausgabebaustein und umgekehrt
- Daten innerhalb des Prozessors von einem Register in ein anderes Register übertragen werden.

Diese Befehle werden in fast jedem Z80-Maschinen- oder Assemblerprogramm benötigt.

Die erste Kategorie der Transferbefehle sind die sogenannten Ladebefehle. Jeder Ladebefehl beginnt mit einem »LD« und einem Leerzeichen. Danach erfolgt die Angabe des Ziels und die Quelle der Daten. Diese beiden Angaben müssen durch ein Komma getrennt werden. Geladen werden kann jedes Register (Ziel) mit dem Inhalt jedes anderen Registers (Quelle) oder einer beliebigen Speicherstelle (Quelle). Es können sogar Register mit ihrem eigenen Inhalt geladen werden, wenn bei Ziel und Quelle die gleiche Registerangabe erfolgt. Die Wirkung eines solchen scheinbar unsinnigen Befehls ist eine kleine Zeitverzögerung des ablaufenden Programms. Diese Verzögerung kann manchmal aber recht nützlich sein (zum Beispiel bei Steuerungen). Die Aussage, daß jedes Register, wie oben angegeben, geladen werden kann, trifft natürlich auch auf den Akkumulator zu. Jedoch besitzt er, wie beim 6502, eine Sonderstellung. Das weiter oben angegebene Beispiel (Lade Akku 6502 und Z80) ist so ein Sonderbefehl für den Akku. Doch zunächst einige Beispiele für normale Transferbefehle:

Lade ein beliebiges Register mit einem angegebenen Wert (unmittelbare Adressierung)

`LD A,1FH` - Bedeutung: Lade den Akkumulator mit dem Wert 1F hexadezimal
`LD B,2AH` - Lade das B-Register mit dem Wert 2A hex.
`LD H,05H` - Lade das H-Register mit dem Wert 5 hex.

Jedes der sogenannten »Arbeitsregister« (A-, B-, C-, D-, E-,H- und L-Register) kann auf diese Weise mit einem beliebigen Wert zwischen 00 und FF (hexadezimal) geladen werden.

Doch nun zum zweiten Beispiel:
Lade ein Register mit dem Inhalt eines anderen Registers.

`LD A,B` - Lade Akkumulator mit dem Inhalt des Registers B
`LD B,C` - Lade das Register B mit dem Inhalt des Registers C
`LD H,A` - Lade das Register H mit dem Inhalt des Akkus

Bereits diese wenigen Beispiele machen das Prinzip deutlich. In gleicher Weise kann aus jedem Arbeitsregister in ein anderes geladen werden. Bei diesen Befehlen wird der Akku wie ein »normales« Register behandelt.

Wie bereits angesprochen, existieren auch Befehle, um die Werte beliebiger Speicherstellen in ein Register zu bringen. Diese Befehle sind nicht sofort durchführbar. Bevor so eine Operation durchgeführt werden kann, muß die Adresse der Speicherstelle in eines der Registerpaare geladen werden.

An dieser Stelle kommen wir erstmals auf die weiter oben bereits erwähnten Registerpaare zu sprechen. Die Kombination aus B- und C-Register wird als BC-Register (16 Bit), die Kombination von D- und E-Register als DE-Register (16 Bit) und die Kombination aus H- und L-Register als HL-Register (16 Bit) bezeichnet. Somit stehen drei »16-Bit-Register« (Arbeitsregister) zur Aufnahme von 16-Bit-Werten, wie zum Beispiel Adressen, zur Verfügung.

Es gibt jetzt zwei Möglichkeiten, diese »16-Bit-Register« mit Werten zu laden. Die erste besteht darin, nacheinander beide Register einzeln mit den beiden Bytes einer Adresse zu laden. Die zweite Möglichkeit besteht in der Verwendung von sogenannten »16-Bit-Befehlen«, mit denen eines der Registerpaare durch einen einzigen Befehl geladen wird.

Soll nun ein beliebiges Register mit dem Inhalt einer Speicherstelle geladen werden, so muß die Adresse der Speicherstelle in einem der Registerpaare stehen. Die Befehlsfolge könnte folgendermaßen aussehen:

`LD H,10H` - Lade das H-Register mit dem Hex-Wert 10
`LD L,00H` - Lade das L-Register mit dem Hex-Wert 00
`LD C,(HL)`- Lade das C-Register mit dem Inhalt der Speicherstelle, deren Adresse im HL-Register steht. In unserem Beispiel wäre das die Adresse 1000 hexadezimal.

Zu beachten ist hierbei, daß das höherwertige Byte der Adresse im H-Register und das niederwertige Byte der Adresse im L-Register stehen muß.

Sehen wir uns noch eine zweite Möglichkeit zum Laden von »16-Bit-Registern« an:

`LD HL,1000H` - Lade das Registerpaar HL mit dem Wert 1000 hex.
`LD C,(HL)` - Die Wirkung ist äquivalent zu obigem Beispiel.
Der Befehl »LD HL,nnnn« ist ein sogenannter 16-Bit-Befehl, das heißt, mit einem einzigen Befehl kann ein 16-Bit-Wert in ein Registerpaar geladen werden. Diese Möglichkeit, die auf alle Registerpaare anwendbar ist (auch auf Programmzähler und Indexregister), erleichtert die Programmierarbeit, spart Zeit und Speicherplatz. Nicht mehr, aber auch nicht weniger leisten diese oft genannten »16-Bit-Befehle« des »8-Bit-Prozessors« Z80.

Wie schon erwähnt, nimmt der Akkumulator unter den Registern eine Sonderstellung ein. So auch beim Laden mit dem Inhalt einer Speicherstelle. Die soeben beschriebenen Ladebefehle sind die allgemeine Form für alle Register. Der Akkumulator kann auch direkt mit dem Inhalt einer beliebigen Speicherstelle geladen werden. Dazu wird der folgende Befehl verwendet:

`LD A,(1000H)` - Lade den Akku mit dem Inhalt der Speicherstelle an Adresse 1000 hex

Bevor wir weitere Befehle (Befehlsgruppen) des Z80 kennenlernen, sei auf eine Besonderheit der meisten Z80-Assembler hingewiesen. Versucht man zum Beispiel mit dem Befehl »LD A,(A000h)« den Inhalt der Speicherstelle A000 (hex) in den Akkumulator zu laden, streiken die meisten Z80-Assembler. Sie erwarten bei Zahlenangaben (Adressen) als erste Ziffer eine Zahl zwischen 0 und 9. Dieses Problem läßt sich auf denkbar einfache Weise lösen. Man fügt zur Zahlenangabe einfach eine führende 0 hinzu.

Der Befehl »LD A,(0A000h)« wird dann anstandslos übersetzt.

Testen Sie Ihren Z80-Assembler in dieser Beziehung, bevor Ihnen unnötigerweise graue Haare wachsen.

Arithmetische Befehle

Der Z80 kann, wie die meisten Mikroprozessoren, grundsätzlich nur addieren und subtrahieren. Da er diese Operationen jedoch sehr schnell ausführt, können andere (Multiplikation oder Division) durch kleine Programme nachgebildet werden. Die Multiplikation 3*4 kann auch als Addition 4+4+4 aufgefaßt und so programmiert werden.

Unterschieden wird bei den arithmetischen Befehlen zwischen Operationen mit Übertrag und ohne Übertrag. Weiterhin ist zu berücksichtigen, daß die meisten Additions- und Subtraktionsbefehle über den Akkumulator laufen. Additionen ohne Berücksichtigung des Übertrags sehen folgendermaßen aus:

`ADD A,03h` - Addiere zum Akkuinhalt den Wert 03 hex. Das Ergebnis steht im Akkumulator.

`ADD A,B` - Addiere zum Akkuinhalt den Inhalt des B-Registers. Das Ergebnis steht im Akku.

`ADD A,(HL)` - Addiere zum Akkuinhalt den Inhalt der Speicherstelle, deren Adresse im HL-Register steht. Das Ergebnis steht im Akku.

Beim letzten Additionsbefehl muß, bevor addiert werden kann, das HL-Register mit der Adresse der Speicherstelle geladen werden (siehe Transferbefehle). Die Doppelregister erlauben es, mit dem Z80 16-Bit-Arithmetik zu programmieren:

`ADD HL,BC` - Addiere zum Inhalt des HL-Registers (16 Bit) den Inhalt des BC-Registers (16 Bit). Das Ergebnis steht im HL-Register.

`SUB 05H` - Subtrahiere vom Akkuinhalt den Wert 5 hex. Das Ergebnis steht im Akkumulator.

`SUB C` - Subtrahiere vom Akkuinhalt den Inhalt des C-Registers. Das Ergebnis steht im Akku.

Soll ein eventuell entstandener Übertrag berücksichtigt werden, programmiert man mit folgenden Befehlen:

`ADC A,09H` - Addiere zum Akkumulatorinhalt die »Hexzahl« 09 und das Übertragsflag C (C = Carryflag). Das Ergebnis steht im Akku.

`ADC A,C` - Addiere zum Akkumulatorinhalt den Inhalt des C-Registers und das Übertragsflag.

Die entsprechenden 16-Bit-Befehle lauten:

`ADC HL,DE` - Addiere zum Inhalt des HL-Registers den Inhalt des DE-Registers und das Übertragsflag C. Das Ergebnis steht im HL-Register.

`SBC A,04H` - Subtrahiere vom Akkuinhalt den Wert 04 hex sowie das Übertragsflag C. Das Ergebnis steht im Akkumulator.

Indizieren mit Hilfe von Registern

Alle Register und Registerpaare können »inkrementiert« oder »dekrementiert« werden. Dabei ist zu beachten, daß die folgenden Befehle, auf Registerpaare angewendet (16 Bit), keine Flags im Statusregister beeinflussen.

```
INC A  ;erhöhe den Inhalt des Akkus um 1
INC B  ;erhöhe den Inhalt des B-Registers um 1
INC HL ;erhöhe den Inhalt des HL-Registerpaares
        um 1
DEC A  ;verringere den Inhalt des Akkus um 1
DEC HL ;verringere den Inhalt des HL-Registers um 1
```

Diese Befehle können für Zählaufgaben oder Zeitschleifen eingesetzt werden.

Weitere arithmetische Befehle finden Sie in der folgenden Tabelle:

DAA	Dezimalanpassung (BCD-Code)
CPL	Bildung des Einerkomplements des Akkus
NEG	Bildung des Zweierkomplements des Akkus

Der Z80 besitzt die Logikbefehle AND (Bitweise UND-Verknüpfung zweier Bytes), OR (Bitweise ODER-Verknüpfung zweier Bytes), XOR (EXKLUSIV-ODER-Verknüpfung zweier Bytes) sowie den Vergleichsbefehl CP (Compare).

`AND 83H` - Führt eine UND-Verknüpfung jedes Bits des Akkus mit jedem Bit der Zahl 83 hex durch. Für den Akkuinhalt 0F hex würde dieser Befehl folgendes bewirken:

Akkuinhalt	0F hex =	00001111 binär
	83 hex =	10000011 binär
AND-Befehl	ergibt :	00000011 binär im Akku

Nach dem AND-Befehl würde also 03 im Akku stehen. Mit Hilfe des AND-Befehls, der auf keinen Fall mit einer Addition verwechselt werden darf, können beliebige Bits »ausgeblendet« werden. In diesem Zusammenhang spricht man auch von »Maskierung«. Diese Maskierung ist bei Ein-/Ausgabeoperationen häufig sehr hilfreich.

Verknüpfungsbefehle des Z80:

AND C	Bitweise UND-Verknüpfung der Inhalte des Akkus und des C-Registers. Ergebnis steht im Akku.
OR 1FH	Bitweise ODER-Verknüpfung des Akkuinhalts mit dem »Datenbyte« 1F hex. Das Ergebnis steht im Akku. Die Verknüpfung wird, wie beim AND-Befehl beschrieben, durchgeführt, mit dem Unterschied, daß, sobald ein Bit im Akku oder im »Datenbyte« oder in beiden den Wert »1« besitzt, das Ergebnis 1 ist.
OR E	Bitweise ODER-Verknüpfung des Akkuinhalts mit dem Inhalt des E-Registers. Das Ergebnis steht im Akku.
XOR 1AH	Bitweise EXCLUSIV-ODER-Verknüpfung des Akkuinhalts mit dem »Datenbyte« 1A hex. Das Ergebnis steht im Akku. Die EXCLUSIV-ODER-Verknüpfung liefert als Ergebnis immer dann eine »1«, wenn die Inhalte der einzelnen Bits unterschiedlich sind. Sind sie gleich, entsteht eine »0«. Dieser Befehl kann zur sogenannten Invertierung (Umkehrung) benutzt werden (bei Logikanpassungen).
CP 1BH	Vergleiche den Akkuinhalt mit dem »Datenbyte« 1B hex. Dabei wird 1Bh vom Akkuinhalt subtrahiert und das Ergebnis nicht weiter berücksichtigt. Es werden lediglich Flags im Statusregister beeinflußt.
CP D	Vergleiche den Akkuinhalt mit dem Inhalt des D-Registers. Die Wirkung ist die gleiche wie bei vorherigem Befehl.

Alle logischen Operationen des Z80 arbeiten nur mit 8-Bit-Daten.

Mit Hilfe der sogenannten Registeranweisungen kann der Inhalt eines Registers verschoben werden. Der Inhalt jedes Bits wird in ein benachbartes Bit verschoben. Beim sogenannten Rotieren wird ein »herausfallendes« Bit jeweils am Anfang oder Ende des Registers verschoben. Beim Z80 kann das Carryflag bei diesen Operationen mit einbezogen werden.

Auch hier wieder ein Beispiel:

`RRA` ;lasse den Inhalt nach rechts rotieren, das heißt jedes Bit (Inhalt) wird um 1 Bit nach rechts verschoben, der Inhalt von Bit 0 gelangt ins Carryflag, der Inhalt des Carryflags in Bit 7.

Weiterhin können mit Hilfe von Registeranweisungen einzelne Bits im Flagregister beeinflußt werden.

Der Z80 ist im Gegensatz zu den meisten anderen Mikroprozessoren mit einer Vielzahl von Befehlen zur Bitmanipulation ausgestattet, deren Beschreibung hier aber zu weit führen würde.

Sprungbefehle werden in fast jedem Maschinenprogramm benötigt. Wie beim 6502 (und anderen Mikroprozessoren) kann ein Sprung, das heißt eine Fortsetzung des Programms an einer angegebenen Adresse, von bestimmten Bedingungen abhängig gemacht werden. Ein Sprung ohne zusätzliche Bedingung wird unbedingter Sprung, ein Sprung, der nur durchgeführt wird, wenn eine zusätzliche Bedingung erfüllt ist, wird bedingter Sprung genannt.

Beispiele für unbedingte Sprünge:

JP 1000h — setze das Programm mit dem Befehl in der Speicherstelle 1000h fort.

JP (HL) setze das Programm an der Adresse fort, die im HL-Register steht.

Die Bedingung, die bei bedingten Sprüngen erfüllt sein muß, damit der jeweilige Sprung durchgeführt wird, ist der Zustand eines entsprechenden Flags im Flag-Register. Es kann der Zustand (0 oder 1) des Z-, C-, P/V- oder S-Flags sein.
Beispiele für bedingte Sprünge:

JP Z,4000h setze das Programm an der Adresse 4000h fort, wenn das Zero-Flag 0 ist.
JP NZ,5000h setze das Programm an der Adresse 5000h fort, wenn das Zero-Flag nicht 0, also 1 ist.

Das Zero-Flag wird auf »1« gesetzt, wenn bei einer arithmetischen oder logischen Operation vorher ein Ergebnis »0« geworden ist.
Beim Z 80 kann eine Programmunterbrechung (Interrupt) durch einen Baustein außerhalb des Prozessors durch entsprechende Befehle zugelassen oder unterbunden werden (maskierbare Interrupts). Hierfür dient der INT-Anschluß am Prozessor.
Manipulation der Interrupts:

EI setze das Interrupt-Flip-Flop. Danach sind Interrupts zugelassen.
DI setze das Interrupt-Flip-Flop zurück. Danach ist ein Interrupt über den INT-Anschluß nicht mehr möglich.

Zusätzlich besitzt der Z80-Mikroprozessor noch eine »nicht maskierbare Interruptleitung« (NMI-Anschluß). Eine Interrupt-Anforderung über diese Leitung wird auf jeden Fall durchgeführt. Weiterhin gibt es beim Z80 noch drei Interruptmodi, die den Ablauf nach einem Interrupt festlegen (Befehle: IM 0, IM 1 und IM 2). Damit der Prozessor bei Unterbrechungen nicht durcheinanderkommt, wird bei einem Interrupt der gerade bearbeitete Befehl zu Ende bearbeitet, bevor der Prozessor auf einen Interrupt reagiert.

Unterprogrammbehandlung

Unterprogramme können beim Z80 mit Hilfe der CALL-Befehle aufgerufen werden. Der CALL-Befehl bewirkt einen Sprung zu einem Unterprogramm an einer angegebenen Startadresse. Wie die Sprungbefehle, können auch die Unterprogrammaufrufe an Bedingungen geknüpft werden. Im Gegensatz zu den Jump-Befehlen (JP) »merkt« sich der Prozessor jedoch seine »Absprungadresse«. Er kann nach der Beendigung des Unterprogramms das ursprüngliche Programm mit dem Befehl fortsetzen, der dem CALL-Befehl im Hauptprogramm folgt. Dieser Rücksprung erfolgt, wenn der Prozessor im Unterprogramm einen RET-Befehl (RETURN) vorfindet, das heißt, zu jedem CALL-Befehl im Hauptprogramm gehört ein RET-Befehl im Unterprogramm. Auch hierzu wieder ein Beispiel:

Hauptprogramm:
beliebiger Befehl
CALL 2000H setze das Programm an der Adresse 2000 hex. fort
nnnn nächster Befehl
Unterprogramm:
2000 beliebiger Befehl
RET springe zurück ins Hauptprogramm zum nächsten Befehl (Adresse nnnn)

Es können nicht nur eigene Unterprogramme aufgerufen werden, sondern auch Unterprogramme des Betriebssystems (CP/M-BDOS-Routinen). Diese Möglichkeit erleichtert die Arbeit mit dem Z80 ganz enorm. In den Programmbeispielen machen wir von dieser Möglichkeit Gebrauch.
In diese Gruppe sind die Befehle eingeordnet, die in keine der anderen Gruppen passen. Zu diesen Befehlen gehört der Befehl »HALT«. Die CPU unterbricht die Programmausführung und führt so lange den Befehl »no operation« (NOP) aus, bis ein Reset durchgeführt wird oder ein Interrupt erfolgt. Der Befehl NOP gehört in diese Gruppe. Er bewirkt nur eine kleine Zeitverzögerung (ein Maschinenzyklus) des ablaufenden Programms. NOPs können in ein Programm genommen werden, um das Programm später an dieser Stelle mit zusätzlichen Befehlen zu erweitern.
Zum Abschluß sollen noch zwei kleine Beispielprogramme (Listing 1 und Listing 2) vorgestellt werden, die für einen Einstieg in die praktische Programmierarbeit benutzt werden können. Um diese Programme auszuprobieren, benötigt man einen Z80-Assembler wie »Double-Ass«, der das mit dem Editor erstellte Assemblerprogramm in den Maschinen-Code übersetzt und einen Linker, der aus dem übersetzten Programm eine direkt unter CP/M aufrufbare »COM«-Datei erstellt.
Die Namenserweiterung muß nach den Erfordernissen des verwendeten Assemblers gewählt werden. Der bei der Erstellung dieser Beispielprogramme verwendete Makroassembler des Small-C-Entwicklungssystems verlangt die Erweiterung MAC, andere Assembler erwarten den Zusatz ASM. Ganze Zeilen werden vom Assembler als Kommentarzeilen angesehen, wenn als erstes Zeichen ein »;« steht.
Die Bedienung Ihres Editors, Assemblers und Linkers entnehmen Sie bitte den Handbüchern zu diesen Programmen. (Rüdiger Szillus/kn)

```
; Testprogramm Z 80 (TEST.MAC)
; Ausgabe eines Zeichens auf dem Bildschirm
;
ORG 0100H       ;Startadresse des Programms
LD C,02H        ;Lade das C-Reg. mit dem Wert 02 hex(BDOS-
                ;Funktion 2)
LD E,"R"        ;Lade das E-Register mit dem ASCII-Wert
                ;für R
CALL 5          ;Sprung ins CP/M-BDOS, Aufruf der BDOS-
                ;Funktion 2
CALL 0          ;Rücksprung zu CP/M
END             ;Ende des Quellprogramms
;
;Beim Sprung ins BDOS wird die BDOS-Funktion
;ausgeführt, deren Nummer im C-Register steht (hier
;BDOS-Funktion 2)
;Die BDOS-Funktion 2 gibt das ASCII-Zeichen auf dem
;Bildschirm
;aus, dessen Wert im E-Register steht (hier R).
```

Listing 1. Routine zur Ausgabe eines einzelnen Zeichens

```
; 2. Testprogramm (TEST2.MAC)
; Ausgabe eines Strings auf dem Bildschirm
;
        ORG 0100h                   ;Startadresse 100 hex.
        LD C,09h                    ;Lade den Wert 09 hex. ins
                                    ;C-Register
        LD DE,TEXT                  ;Lade die Anfangsadresse
                                    ;von "TEXT:"
        CALL 5                      ;Sprung ins BDOS
        CALL 0                      ;Rücksprung zu CP/M
;
TEXT:   DB "Rüdiger Szillus $";Definition des Datenbytes
        END                         ;Ende des Quellprogramms
;
;Beim Sprung ins BDOS wird die BDOS-Funktion 9
;ausgeführt.
;(Nummer steht im C-Register).
;Die Adresse des auszugebenden Strings steht im DE-
;Register ;(hier Label TEXT:).
;BDOS-Funktion 9 = Ausgabe eines Strings auf dem
;Bildschirm.
;Das $-Zeichen markiert das Ende des Strings und muss
;angegeben werden.
```

Listing 2. Routine zur Ausgabe eines Strings

Weltrekord: Disketten-Backup in 8 Sekunden



Da werden sogar die Besitzer von Personal Computern neidisch. »Double-Touch« kopiert mit zwei Laufwerken eine Diskette in sage und schreibe 8 Sekunden.

Die vielfältigen Möglichkeiten bei der Programmierung der Floppy 1570/71 fordern regelrecht dazu heraus, schnelle Kopierprogramme für diese Laufwerkstypen zu schreiben. Ganz besonders interessant wird es, wenn neben dem C 128 gleich zwei Laufwerke zur Verfügung stehen. Dann kann man sich nicht nur das lästige und zeitraubende Diskettenwechseln ersparen, sondern erzielt auch eine enorme Geschwindigkeit. »Double-Touch« (Listing 1) kopiert eine Diskette nämlich in rund 8 (!) Sekunden und formatiert sie auch noch gleichzeitig. Wünschen Sie die Verifizierung der Kopie, so werden weitere 8 Sekunden benötigt, die jedoch jeder Anwender aus Gründen der Datensicherheit opfern sollte.

Verwendet man zwei 1571-Laufwerke, so können auch doppelseitig bespielte Disketten (aber keine doppelseitigen 1541-Formate!) bei entsprechend verdoppeltem Zeitaufwand in einem Durchgang kopiert werden.

Komfortable Menüsteuerung

»Gemischter« Betrieb mit einer 1570- und einer 1571-Floppy ist natürlich auch möglich; man ist dann, wie bei Verwendung zweier 1570-Laufwerke, auf einseitige Disketten beschränkt. Mit der Floppy 1541 arbeitet »Double-Touch« allerdings nicht zusammen.

Bevor Sie jedoch in den Genuß dieser großartigen Leistungsmerkmale kommen, müssen Sie noch Listing 1 mit dem MSE (im C 64-Modus) eingeben. Danach wird das Programm im C 128-Modus mit RUN"DOUBLE-TOUCH" geladen und zugleich gestartet, worauf das Hauptmenü, wie oben sichtbar, auf dem 80-Zeichen-Bildschirm dargestellt wird.

Der erste der vier Menüpunkte, »DIRECTORY ANZEIGEN«, gibt das Inhaltsverzeichnis aus, wahlweise von Laufwerk 8 oder 9. Das Directory wird, um die Initialisierung bei einseitigen Disketten abzukürzen, im 1541-Modus eingelesen. Dadurch wird bei doppelseitigen Disketten allerdings ein falscher Wert als Anzahl der freien Blöcke angezeigt.

Mit »BACKUP DISK« kopiert man eine Diskette. Bei den drei Eingabeaufforderungen, die nun folgen, sind bereits Standardwerte vorgegeben, die mit <RETURN> übernommen werden können.

Kopierparameter einstellen

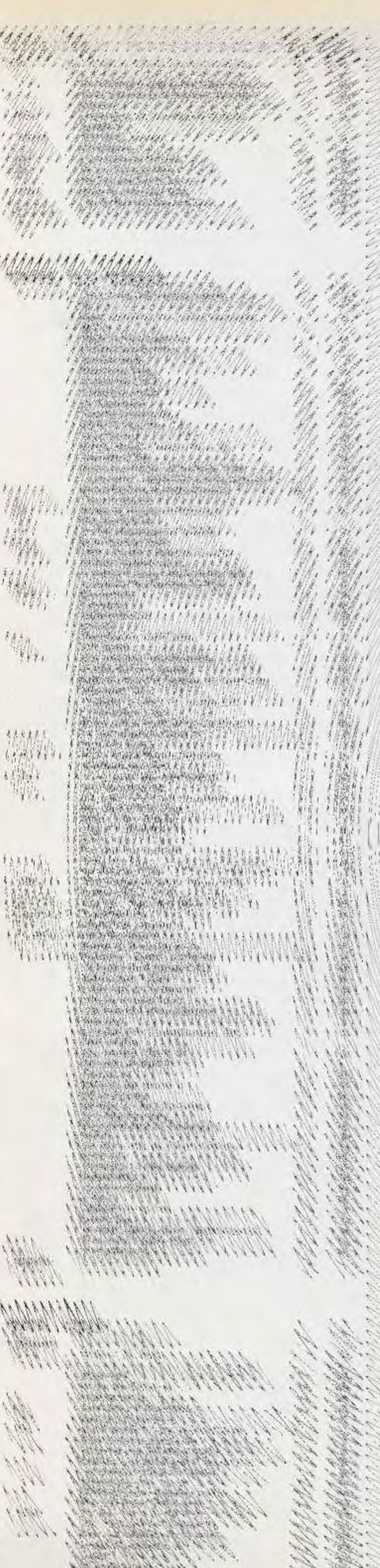
»SINGLE- ODER DOUBLE-SIDED« wählt das Diskettenformat, wobei beim Arbeiten mit einer 1570-Floppy nur »SINGLE-SIDED« in Frage kommt.

Die zweite Frage, »ZIELDISKETTE VERIFIZIEREN?«, sollte üblicherweise mit »J« beantwortet werden, um eine fehlerfreie Kopie zu gewährleisten. Die Frage »GERAETE-ADRESSE AENDERN« muß nur dann bejaht werden, wenn die beiden verwendeten Laufwerke auf Geräteadresse 8 eingestellt sind. Dann wird vor dem Kopiervorgang die Adresse eines Laufwerks softwaremäßig von 8 auf 9 geändert. Dies muß allerdings vor jedem Kopiervorgang geschehen. Eine hardwaremäßige Einstellung der Geräteadressen ist daher zu empfehlen.

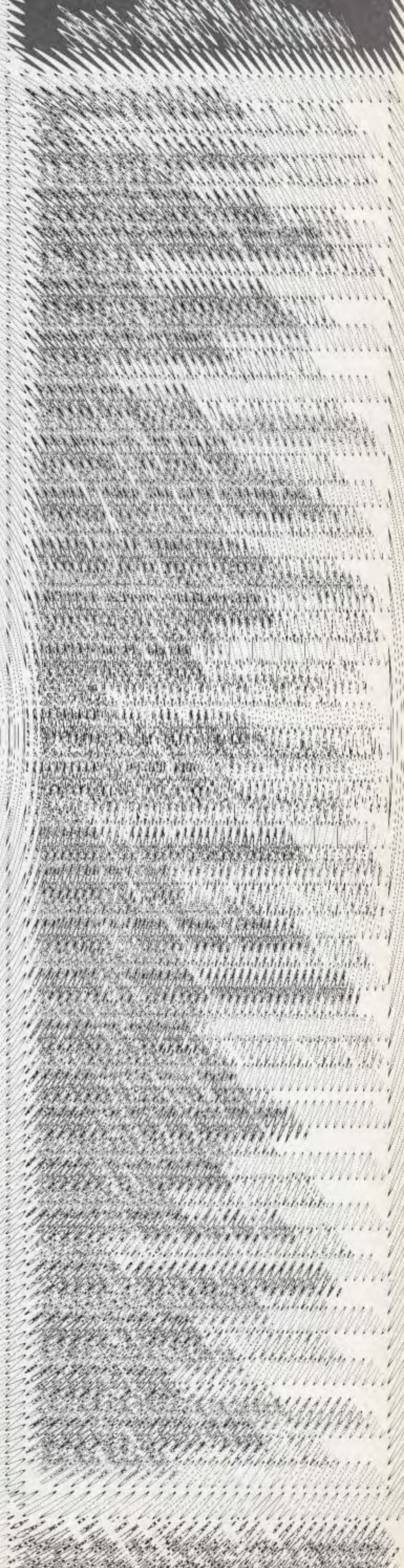
Nun beginnt der eigentliche Kopiervorgang. Es erfolgt nacheinander die Aufforderung, Quell- und Zieldiskette einzulegen. Die LEDs (Leuchtdioden) des Laufwerks zeigen jeweils an, wo die Disketten einzulegen sind, eine Verwechslung dürfte daher schwer fallen.

Während des Kopierens können auch Fehlermeldungen auftreten. »WRITE PROTECT ON« bedeutet, daß sich ein Schreibschutz auf der eingelegten Diskette befindet oder daß noch keine Schreibkerbe angebracht wurde und führt zum sofortigen Abbruch des Kopiervorgangs. »x SECTORS MISSING ON TRACK y« zeigt an, daß die Anzahl der einge-









lesenen Sektoren einer Spur auf Diskette nicht stimmt; Ursache hierfür ist ein Read-Error 20 oder 21. Andere Lesefehler (Read-Error 22, 23, 24, 27 und 29) werden wie die beiden ersten mitkopiert, ohne jedoch erkannt zu werden. Ein Kopieren von kopiergeschützten Disketten ist trotzdem nicht empfehlenswert, da bestimmte Fehler der Quelldiskette, die normalerweise als »READ ERROR 21« angegeben werden, zur Zerstörung weiterer Sektoren führen können.

So werden Fehler behandelt

Weiteren Aufschluß über Fehler erhält man nur bei angewähltem Verifizieren der Zieldiskette. Hier wird versucht, die kopierte Spur einzulesen. Sektoren, bei denen dies nicht einwandfrei gelingt, werden mit »VERIFY ERROR« angezeigt. Tritt ein solcher Fehler auf, so kann dies auch an einer fehlerhaften oder kopiergeschützten Quelldiskette liegen. Da »Double-Touch« nur einen Leseversuch vornimmt, empfiehlt es sich, in einem solchen Fall ein anderes - langsameres - Kopierprogramm, das mehrere Leseversuche durchführt, oder einen Nibbler zu verwenden. Nun ja, Schnelligkeit hat eben seinen Preis!

Ein seltener Fehler ist »BUFFER OVERFLOW ON TRACK x«. Manchmal ist ein Read-Error 21 die Ursache, aber auch bei fehlerfreien Disketten kann dieser Fehler auftreten. Das Ergebnis ist jedoch - wie oben - immer dasselbe: Die betroffene Spur wurde nicht richtig kopiert und einige oder alle Sektoren des Tracks sind daher fehlerhaft.

Als Abhilfe kann man bei Problemdisketten zunächst versuchen, die Diskette nochmals zu kopieren. Tritt der Fehler wieder auf, hilft möglicherweise (beispielsweise, wenn die Köpfe beider Laufwerke leicht unterschiedlich justiert sind) der Hauptmenü-Punkt »QUELLE UND ZIEL VERTAUSCHEN«, womit man das Laufwerk mit der Geräteadresse 9 als Quell-Laufwerk definiert (bitte auch Disketten austauschen). Entsteht der Fehler auch bei erneutem Versuch, so muß man sich wohl oder übel damit abfinden, daß sich die betroffene Diskette mit »Double-Touch« nicht kopieren läßt. In der Regel ist die Anzahl solch hartnäckiger Disketten aber so verschwindend gering, daß dies keinen Nachteil darstellt. In der 64'er-Redaktion wird »Double-Touch« schon seit einigen Monaten intensiv verwendet, und es sind bislang noch keine Probleme aufgetreten.

Das Hauptmenü hält schließlich noch einen weiteren Menüpunkt bereit: Nach »QUIT« findet man sich kurzerhand im C128-Modus wieder; sollte das Diskettenlaufwerk anschließend nicht ansprechbar sein, hilft ein Hardware-Reset (Reset-Schalter betätigen) oder kurzzeitiges Abschalten des Diskettenlaufwerks weiter.

Zum Schluß sei noch erwähnt, daß man durch Drücken von <RESTORE> jederzeit ins Hauptmenü gelangt. Geschieht dies mitten in einem Kopiervorgang, so beenden die Laufwerke unverzüglich ihre Tätigkeit.

(Reiner Koukal/Florian Müller)

```
NAME : DOUBLE TOUCH        1C01 2D37
------------------------------------
1C01 : 16 1C 00 00 FE 02 20 31  08
1C09 : 35 3A DE 20 9C 3A 9E 20  6D
1C11 : 37 32 31 32 00 00 00 20  34
1C19 : 46 55 45 52 20 56 49 43  06
1C21 : 4B 59 20 55 4E 44 20 54  FC
1C29 : 4F 4D 20 A9 08 85 26 A9  F5
1C31 : 09 85 27 A9 00 8D 00 FF  68
1C39 : A9 1F A2 52 8D 19 03 8E  30
1C41 : 18 03 BA 86 20 20 84 FF  6F
1C49 : 20 81 FF 24 D7 30 03 20  F9
1C51 : 2E CD A9 C0 85 F7 A9 80  A8
1C59 : 85 F8 20 B6 77 A9 00 20  3E
1C61 : 79 1F 20 45 E5 20 44 1F  C9
1C69 : A9 0A 20 79 1F A9 01 20  D2
1C71 : 79 1F A9 07 20 79 1F 20  50
1C79 : C2 1F A2 04 DD 32 25 F0  DA
1C81 : 05 CA D0 F8 F0 F1 8A 48  98
1C89 : 20 44 1F 68 0A AA BD 36  FA
1C91 : 25 48 BD 35 25 48 60 AD  62
1C99 : DA 25 D0 0B A9 10 20 79  2A
1CA1 : 1F 20 C2 1F 4C 63 1C A9  08
1CA9 : 03 20 79 1F 6C FC FF A9  01
1CB1 : 02 20 79 1F 20 81 20 A2  D9
1CB9 : 08 C9 0D D0 02 A9 38 C9  E5
1CC1 : 38 F0 06 A2 09 C9 39 D0  AD
1CC9 : EB 86 21 20 87 20 20 E3  05
1CD1 : 1F 20 27 1F A9 05 20 E0  B3
1CD9 : 1F A9 00 85 90 A5 21 20  79
1CE1 : B1 FF A9 F0 20 93 FF A9  0C
1CE9 : 24 20 A8 FF 20 AE FF A5  0A
1CF1 : 21 20 B4 FF A9 60 20 96  9B
1CF9 : FF 20 A5 FF 20 A5 FF 20  E1
1D01 : 43 1D A9 03 20 E0 1F A9  76
1D09 : 0C 20 E0 1F A9 10 85 1E  AF
1D11 : 20 43 1D C6 1E D0 F9 A9  97
1D19 : 0F 20 79 1F 20 C2 1F C9  A3
1D21 : 20 F0 DF 20 33 1D 4C 63  C9
1D29 : 1C 20 33 1D 20 BD 1F 4C  CB
1D31 : 63 1C 20 AB FF A5 21 20  12
1D39 : B1 FF A9 E0 20 93 FF 4C  A7
1D41 : AE FF A2 03 20 74 1D B0  73
1D49 : 25 CA D0 F8 AA 20 74 1D  DE
1D51 : B0 1C 20 E8 1F A9 20 20  34
1D59 : 2D C7 20 74 1D B0 0F AA  E9
1D61 : F0 06 20 2D C7 4C 5B 1D  89
1D69 : A9 0D 20 2D C7 60 68 68  39
1D71 : 4C 2A 1D 20 A5 FF 38 A4  A2
1D79 : 90 D0 01 18 60 A5 26 A2  C6
1D81 : 08 A0 09 C9 09 F0 04 A2  C2
1D89 : 09 A0 08 86 26 84 27 A9  2C

1D91 : 12 20 79 1F A5 26 18 69  B4
1D99 : 30 20 2D C7 A9 13 20 79  C4
1DA1 : 1F A5 27 18 69 30 [illegible] 2D  53
1DA9 : C7 20 C2 1F 4C 63 1C A9  B8
1DB1 : 04 20 79 1F 20 81 20 A2  DB
1DB9 : 23 C9 0D D0 02 A9 53 C9  6D
1DC1 : 53 F0 06 A2 46 C9 44 D0  C8
1DC9 : EB 8E A3 2B 8E A3 26 86  F5
1DD1 : 23 20 87 20 20 2D C7 A9  C8
1DD9 : 01 85 18 A2 4A A9 06 20  42
1DE1 : 59 1F 8E A5 26 A2 4E A9  26
1DE9 : 14 20 59 1F 8A 10 25 A9  59
1DF1 : 0E 20 79 1F 20 C2 1F A9  39
1DF9 : 08 85 21 A9 25 A2 CF 20  28
1E01 : 2B 1F A9 25 A2 D5 20 2B  7A
1E09 : 1F A9 11 20 79 1F 20 C2  DC
1E11 : 1F 20 E3 1F A9 08 85 21  50
1E19 : A9 25 A2 CF 20 2B 1F A9  23
1E21 : 09 85 21 A9 25 A2 CF 20  51
1E29 : 2B 1F A9 02 20 E0 1F A9  67
1E31 : 2B A2 00 A4 26 20 F8 1F  C7
1E39 : A9 26 A2 00 A4 27 20 F8  94
1E41 : 1F A0 09 84 21 A9 25 A2  BC
1E49 : 65 20 2B 1F A0 08 84 21  0C
1E51 : 20 4B 20 78 20 45 E5 20  32
1E59 : 51 21 20 82 21 A9 2B A0  E1
1E61 : 00 A2 05 20 29 21 20 4E  B0
1E69 : E5 A9 26 A0 00 A2 05 20  2A
1E71 : 29 21 20 45 E5 58 A9 0B  B9
1E79 : 20 79 1F 20 C2 1F 20 4E  64
1E81 : E5 A9 0C 20 79 1F 20 C2  D9
1E89 : 1F 20 E3 1F 78 EE DA 25  4A
1E91 : 20 3D 21 20 57 E5 20 45  4C
1E99 : E5 20 1C 1F A9 0D 20 79  F0
1EA1 : 1F A9 09 20 E0 1F 20 91  86
1EA9 : 20 A9 00 85 24 85 25 20  92
1EB1 : 4E E5 A2 0A CA D0 FD 20  47
1EB9 : 45 E5 2C 00 DD 50 FB 20  8C
1EC1 : 51 21 20 3D 21 20 6C 21  5A
1EC9 : C9 FE D0 09 58 20 44 1F  3C
1ED1 : A9 15 4C 13 1F C9 FF D0  5C
1ED9 : 07 A9 01 85 25 4C C6 1E  B2
1EE1 : 29 FF 30 05 85 24 4C C6  EF
1EE9 : 1E 29 1F F0 0D AA 20 6C  01
1EF1 : 21 A8 A9 01 99 FF 0A CA  48
1EF9 : D0 F4 20 9E 20 E6 18 C6  46
1F01 : 23 F0 0C A5 18 C9 24 D0  56
1F09 : 03 20 1C 1F 4C A7 1E 58  33
1F11 : A9 05 20 79 1F 20 C2 1F  B0
1F19 : 4C 63 1C 2C 00 DD 70 FB  4C
1F21 : 2C 00 DD 50 FB 60 A9 25  82
1F29 : A2 5F 8D 38 1F 8E 37 1F  67

1F31 : 20 63 20 A2 00 BD FF FF  4D
1F39 : F0 06 20 A8 FF E8 D0 F5  C0
1F41 : 4C AE FF A9 04 85 1E A5  4A
1F49 : 1E 20 E0 1F C6 1E D0 F7  24
1F51 : 60 A6 20 9A 58 4C 34 1C  51
1F59 : 86 1E 20 79 1F 20 81 20  5F
1F61 : A2 FF C9 0D D0 02 A5 1E  07
1F69 : C9 4A F0 06 A2 00 C9 4E  42
1F71 : D0 EB 20 87 20 4C 2D C7  D9
1F79 : 48 AA BD E1 24 20 E0 1F  C7
1F81 : A9 21 A2 8D 85 1B 86 1A  95
1F89 : 68 AA F0 0A A0 00 20 B4  B8
1F91 : 1F D0 FB CA D0 F8 A0 00  C8
1F99 : B1 1A A2 01 C9 23 D0 07  27
1FA1 : 20 B4 1F AA 20 B4 1F 20  9D
1FA9 : 2D C7 CA D0 FA 20 B4 1F  48
1FB1 : D0 E8 60 E6 1A D0 02 E6  E8
1FB9 : 1B B1 1A 60 A9 08 20 79  8E
1FC1 : 1F A9 00 85 D0 85 D1 20  26
1FC9 : EF EE F0 FB 60 0A 0A AA  BF
1FD1 : A0 00 BD F7 24 99 E4 00  82
1FD9 : E8 C8 C0 04 D0 F4 60 20  4C
1FE1 : CE 1F A9 93 4C 2D C7 85  74
1FE9 : 61 86 60 20 07 BA A9 00  96
1FF1 : A2 08 A0 03 4C 5D BA 85  C5
1FF9 : 1B 86 1A 84 21 A9 03 A2  1F
2001 : 00 85 1D 86 1C A2 05 20  07
2009 : 63 20 20 6D 20 A9 57 20  1F
2011 : A8 FF A5 1C 20 A8 FF A5  38
2019 : 1D 20 A8 FF A9 20 20 A8  DE
2021 : FF A0 00 B1 1A 20 A8 FF  EC
2029 : C8 C0 20 D0 F6 20 AE FF  9F
2031 : A5 1A 18 69 20 85 1A 90  CE
2039 : 02 E6 1B A5 1C 18 69 20  92
2041 : 85 1C 90 02 E6 1D CA D0  5D
2049 : BE 60 20 63 20 20 6D 20  A5
2051 : A9 45 20 A8 FF A9 00 20  47
2059 : A8 FF A9 03 20 A8 FF 4C  AB
2061 : AE FF A5 21 20 B1 FF A9  7F
2069 : 6F 4C 93 FF A9 4D 20 A8  BA
2071 : FF A9 2D 4C A8 FF A5 21  7D
2079 : 20 B4 FF A9 6F 4C 96 FF  DC
2081 : 20 73 CD 4C C2 1F 48 8A  B3
2089 : 48 20 A3 CD 68 AA 68 60  C2
2091 : A9 00 A2 14 9D 00 0B CA  01
2099 : 10 FA 85 24 60 A9 00 85  6B
20A1 : F8 A6 24 F0 1F A9 25 A2  2D
20A9 : CB 20 10 21 A6 24 A9 00  DF
20B1 : 20 E8 1F A9 25 A2 8C 20  1C
20B9 : 10 21 A9 00 A6 18 20 E8  42
20C1 : 1F 20 24 21 A5 25 F0 13  8B
20C9 : A9 25 A2 A7 20 10 21 A6  F7
```

```
20D1 : 18 20 E8 1F A9 25 A2 C5   F1
20D9 : 20 10 21 A0 00 84 1E B9   6E
20E1 : 00 0B F0 1F A9 25 A2 6A   AA
20E9 : 20 10 21 A6 18 A9 00 20   3E
20F1 : E8 1F A9 25 A2 81 20 10   4F
20F9 : 21 A6 1E A9 00 20 E8 1F   0D
2101 : 20 24 21 A4 1E C8 C0 15   66
2109 : D0 D3 A9 80 85 F8 60 A0   20
2111 : 00 8D 1A 21 8E 19 21 B9   2C
2119 : FF FF F0 06 20 2D C7 C8   31
2121 : D0 F5 60 A9 0D 4C 2D C7   B1
2129 : 84 1A 85 1B A0 A0 B1 1A   89
2131 : 20 77 21 C8 D0 F8 E6 1B   15
2139 : CA D0 F3 60 AD 05 D5 29   21
2141 : F7 8D 05 D5 AD 0E DC 29   0C
2149 : BF 8D 0E DC 2C 0D DC 60   4D
2151 : AD 05 D5 09 08 8D 05 D5   C4
2159 : A2 01 8E 04 DC A2 00 8E   A0
2161 : 05 DC A9 41 8D 0E DC 2C   7C
2169 : 0D DC 60 A9 08 2C 0D DC   01
2171 : F0 FB AD 0C DC 60 8D 0C   6B
2179 : DC A9 08 2C 0D DC F0 FB   25
2181 : 60 A2 14 A0 00 88 D0 FD   CF
2189 : CA D0 FA 60 8E 1E 20 D5   8C
2191 : 23 4C C0 C9 0D 20 DD 20   CD
2199 : 20 9E 44 4F 55 42 4C 45   26
21A1 : 2D 54 4F 55 43 48 20 46   FA
21A9 : 55 45 52 20 5A 57 45 49   41
21B1 : 20 31 35 37 30 2F 37 31   5A
21B9 : 2D 46 4C 4F 50 50 49 45   3E
21C1 : 53 20 20 20 20 28 43 29   D3
21C9 : 31 39 38 36 20 42 59 20   25
21D1 : 52 45 49 4E 45 52 20 4B   E0
21D9 : 4F 55 4B 41 4C 20 4A 55   67
21E1 : 4E 2E 1E 20 20 DD 0D 20   37
21E9 : CA 23 4C C0 CB 0D 20 99   49
21F1 : D5 23 4C C0 C9 0D 20 DD   C4
21F9 : 23 4C 20 DD 0D 20 CA 23   49
2201 : 4C C0 CB 0D 23 10 11 20   79
2209 : D5 23 4C C0 C9 0D 20 DD   DC
2211 : 23 4C 20 DD 0D 20 CA 23   61
2219 : 4C C0 CB 05 00 11 11 11   48
2221 : 11 5B 44 5D 20 20 44 49   43
2229 : 52 45 43 54 4F 52 59 20   A6
2231 : 41 4E 5A 45 49 47 45 4E   59
2239 : 11 0D 5B 42 5D 20 20 42   CC
2241 : 41 43 4B 55 50 20 44 49   4B
2249 : 53 4B 11 0D 5B 43 5D 20   AD
2251 : 20 51 55 45 4C 4C 45 20   94
2259 : 55 4E 44 20 5A 49 45 4C   88
2261 : 20 56 45 52 54 41 55 53   93
2269 : 43 48 45 4E 11 0D 5B 51   75
2271 : 5D 20 20 51 55 49 54 20   42
2279 : 2D 20 50 52 4F 47 52 41   10
2281 : 4D 4D 20 42 45 45 4E 44   05
2289 : 45 4E 11 00 23 14 20 44   16
2291 : 49 52 45 43 54 4F 52 59   79
2299 : 20 56 4F 4E 20 4C 41 55   96
22A1 : 46 57 45 52 4B 20 38 20   05
22A9 : 4F 44 45 52 20 39 20 3F   81
22B1 : 20 38 9D 00 23 0B 11 23   6A
22B9 : 22 20 54 53 43 48 55 45   C1
22C1 : 53 20 2E 2E 2E 00 53 49   38
22C9 : 4E 47 4C 45 20 4F 44 45   8F
22D1 : 52 20 44 4F 55 42 4C 45   51
22D9 : 20 53 49 44 45 44 20 20   B5
22E1 : 3F 20 20 53 9D 00 23 1A   3D
22E9 : 20 4B 4F 50 49 45 52 56   41
22F1 : 4F 52 47 41 4E 47 20 42   87
22F9 : 45 45 4E 44 45 54 20 21   B7
2301 : 00 5A 49 45 4C 44 49 53   DC
2309 : 4B 45 54 54 45 20 56 45   D0
2311 : 52 49 46 49 5A 49 45 52   6C
2319 : 45 4E 20 3F 20 20 4A 9D   DD
2321 : 00 23 17 20 45 4E 54 53   3B
2329 : 50 52 45 43 48 45 4E 44   CD
2331 : 45 20 54 41 53 54 45 20   F1
2339 : 44 52 55 45 43 4B 45 4E   E5
2341 : 20 21 00 23 1B 20 42 49   A5
2349 : 54 54 45 20 54 41 53 54   62
2351 : 45 20 44 52 55 45 43 4B   25
2359 : 45 4E 20 21 00 00 23 1D   B8
2361 : 20 42 49 54 54 45 20 57   1E
2369 : 41 45 48 4C 45 4E 20 53   D6
2371 : 49 45 3A 00 23 1A 20 51   12
2379 : 55 45 4C 4C 44 49 53 4B   80
2381 : 45 54 54 45 20 45 49 4E   9C
2389 : 4C 45 47 45 4E 20 21 00   5D
2391 : 23 1A 20 5A 49 45 4C 44   8D
2399 : 49 53 4B 45 54 54 45 20   45
23A1 : 45 49 4E 4C 45 47 45 4E   E8
23A9 : 20 21 00 23 1C 20 53 43   55
23B1 : 48 52 45 49 42 2D 4C 45   E6
23B9 : 53 45 2D 46 45 48 4C 45   15
```

```
23C1 : 52 00 23 0D 20 4E 55 52   EC
23C9 : 20 45 49 4E 20 4C 41 55   BC
23D1 : 46 57 45 52 4B 20 45 49   BC
23D9 : 4E 53 43 48 41 4C 54 45   FD
23E1 : 4E 20 55 4E 44 20 54 41   78
23E9 : 53 54 45 20 44 52 55 45   72
23F1 : 43 4B 45 4E 20 21 00 23   46
23F9 : 14 20 57 45 49 54 45 52   8D
2401 : 20 4D 49 54 20 53 50 41   05
2409 : 43 45 20 2D 20 45 4E 44   8B
2411 : 45 20 4D 49 54 20 54 41   FD
2419 : 53 54 45 20 21 00 23 12   AF
2421 : 20 49 43 48 20 48 41 42   8E
2429 : 45 20 44 4F 43 48 20 4E   0D
2431 : 4F 43 48 20 47 41 52 20   40
2439 : 4E 49 43 48 54 53 20 47   F5
2441 : 45 4D 41 43 48 54 20 21   D0
2449 : 00 23 12 20 5A 57 45 49   6B
2451 : 54 45 53 20 4C 41 55 46   D1
2459 : 57 45 52 4B 20 41 55 43   39
2461 : 48 20 45 49 4E 53 43 48   51
2469 : 41 4C 54 45 4E 20 21 00   F9
2471 : 51 55 45 4C 4C 41 55 46   F8
2479 : 57 45 52 4B 20 2D 20 00   5D
2481 : 5A 49 45 4C 4C 41 55 46   0B
2489 : 57 45 52 4B 20 2D 20 00   6D
2491 : 47 45 52 41 45 54 45 41   C6
2499 : 44 52 45 53 53 45 20 41   24
24A1 : 45 4E 44 45 52 4E 20 20   1F
24A9 : 20 20 3F 20 20 4E 9D 00   98
24B1 : 23 10 20 57 52 49 54 45   1B
24B9 : 20 50 52 4F 54 45 43 54   A5
24C1 : 20 4F 4E 20 2D 20 53 43   C8
24C9 : 48 52 45 49 42 53 43 48   11
24D1 : 55 54 5A 20 45 4E 54 46   90
24D9 : 45 52 4E 45 4E 20 21 00   EE
24E1 : 00 04 03 00 06 03 08 03   43
24E9 : 03 09 01 03 03 01 03 03   5C
24F1 : 03 03 0D 0E 07 03 18 00   64
24F9 : 00 4F 04 04 02 4D 15 06   0D
2501 : 00 4F 17 17 02 4D 15 06   3D
2509 : 18 4F 04 04 1A 4D 0B 0B   99
2511 : 18 4F 0F 0F 18 4F 0D 0D   C1
2519 : 18 4F 15 06 16 4F 00 00   BB
2521 : 00 00 00 00 00 00 [illegible] 06   82
2529 : 1A 4F 17 17 16 4D 17 17   EA
2531 : 29 4D 44 42 43 51 AF 1C   10
2539 : AF 1D 7D 1D 97 1C 2D 2D   E3
2541 : 2D 20 47 41 4E 44 41 4C   1D
2549 : 46 20 46 4F 52 20 50 52   27
2551 : 45 53 49 44 45 4E 54 20   73
2559 : 21 20 2D 2D 2D 00 55 30   04
2561 : BE 4D 30 00 55 30 BE 08   B4
2569 : 00 56 45 52 49 46 59 2D   B6
2571 : 45 52 52 4F 52 20 4F 4E   5E
2579 : 20 54 52 41 43 4B 20 00   8F
2581 : 20 2D 20 53 45 43 54 4F   09
2589 : 52 20 00 20 53 45 43 54   05
2591 : 4F 52 53 20 4D 49 53 53   F5
2599 : 49 4E 47 20 4F 4E 20 54   70
25A1 : 52 41 43 4B 20 00 2A 2A   CD
25A9 : 2A 20 42 55 46 46 45 52   6F
25B1 : 20 4F 56 45 52 46 4C 4F   DE
25B9 : 57 20 4F 4E 20 54 52 41   2E
25C1 : 43 4B 20 00 20 2A 2A 2A   02
25C9 : 0D 00 20 20 20 00 55 30   9A
25D1 : BE 4D 31 00 55 30 BE 09   66
25D9 : 00 00 00 00 00 00 00 00   DA
25E1 : 00 00 00 00 00 00 00 00   E2
25E9 : 00 00 00 00 00 00 00 00   EA
25F1 : 00 00 00 00 00 00 00 00   F2
25F9 : 00 00 00 00 00 00 00 78   EA
2601 : A9 00 8D 00 18 20 7C 87   91
2609 : 20 64 87 AD 0F 18 09 20   09
2611 : 8D 0F 18 AD 0C 1C 09 0E   C4
2619 : 8D 0C 1C 20 CE 81 20 B2   96
2621 : 81 A9 04 2C 00 18 F0 FB   7A
2629 : 2C 0D 40 A0 A0 A2 05 A9   86
2631 : 08 2C 0D 40 F0 FB AD 0C   58
2639 : 40 99 00 03 C8 D0 F0 EE   5B
2641 : 3C 03 CA D0 EA A9 04 2C   30
2649 : 00 18 D0 FB 2C 00 18 F0   0E
2651 : FB A2 FE AD 64 03 49 A0   D7
2659 : 8D 64 03 20 6C 03 CA D0   89
2661 : FA E8 E0 FE F0 ED 20 6C   BF
2669 : 03 D0 F6 8A 48 49 FF A8   03
2671 : 20 7C 87 A9 04 2C 00 18   B8
2679 : F0 13 CA D0 F6 20 88 87   61
2681 : A9 04 2C 00 18 F0 06 88   6A
2689 : D0 F6 68 AA 60 20 7C 87   4C
2691 : 2C 0D 40 4C A0 03 00 00   00
2699 : 00 00 00 00 00 00 00 4C   32
26A1 : A6 03 46 00 00 A9 06 8D   DB
26A9 : 04 40 A9 00 8D 05 40 A9   8D
```

```
26B1 : 11 8D 0E 40 20 B2 81 EE   90
26B9 : A4 03 CE A3 03 10 09 20   1C
26C1 : AE E9 20 88 87 6C FC FF   4D
26C9 : AD A4 03 C9 01 F0 04 C9   FE
26D1 : 24 D0 06 20 C2 05 4C FF   68
26D9 : 03 A9 03 85 86 A0 00 A2   D5
26E1 : 00 A9 04 2C 00 18 D0 13   67
26E9 : CA D0 F8 88 D0 F5 C6 86   4F
26F1 : D0 F1 20 AE E9 20 B7 E9   EA
26F9 : 4C 51 05 20 13 06 AD A4   95
2701 : 03 C9 24 30 03 38 E9 23   D8
2709 : AA BD 2B 94 85 82 85 83   79
2711 : AD 00 1C 29 9F 1D 08 94   17
2719 : 8D 00 1C A9 00 85 88 20   71
2721 : 38 06 AD 00 1C 29 10 D0   B5
2729 : 03 4C 51 05 A9 FF 8D 03   1E
2731 : 1C AD 0C 1C 29 1F 09 C0   DC
2739 : 8D 0C 1C A9 55 8D 01 1C   07
2741 : A2 FF 9A A0 FF A9 04 2C   53
2749 : 00 18 D0 03 4C 55 05 A9   C1
2751 : 08 2C 0D 40 F0 EF AD 0C   18
2759 : 40 48 BA 30 E8 A2 05 A9   7D
2761 : 08 2C 0D 40 F0 04 AD 0C   C9
2769 : 40 48 2C 0F 18 30 F0 A9   D4
2771 : FF 8D 01 1C CA D0 E8 A2   17
2779 : 0A A9 08 2C 0D 40 F0 04   7E
2781 : AD 0C 40 48 2C 0F 18 30   49
2789 : F0 B9 00 01 8D 01 1C 88   D8
2791 : CA D0 E6 A2 07 A9 08 2C   07
2799 : 0D 40 F0 04 AD 0C 40 48   50
27A1 : 2C 0F 18 30 F0 A9 55 8D   2E
27A9 : 01 1C CA D0 E8 A2 05 A9   90
27B1 : 08 2C 0D 40 F0 04 AD 0C   19
27B9 : 40 48 2C 0F 18 30 F0 A9   24
27C1 : FF 8D 01 1C CA D0 E8 A2   67
27C9 : 45 A9 08 2C 0D 40 F0 04   09
27D1 : AD 0C 40 48 2C 0F 18 30   99
27D9 : F0 B9 00 01 8D 01 1C 88   28
27E1 : CA D0 E6 A9 08 2C 0D 40   99
27E9 : F0 04 AD 0C 40 48 2C 0F   DD
27F1 : 18 30 F0 B9 00 01 8D 01   D5
27F9 : 1C 88 CA D0 E6 A2 08 A9   1D
2801 : 08 2C 0D 40 F0 04 AD 0C   69
2809 : 40 48 2C 0F 18 30 F0 A9   74
2811 : 55 8D 01 1C CA D0 E8 C6   55
2819 : 82 F0 39 BA 86 86 A9 08   0C
2821 : 2C 0D 40 F0 04 AD 0C 40   60
2829 : 48 98 38 E5 86 30 1E C9   7E
2831 : 0D 10 1A AA A9 08 2C 0D   C8
2839 : 40 F0 09 AD 0C 40 48 E8   9F
2841 : E0 0D 10 09 A9 04 2C 00   39
2849 : 18 D0 E9 F0 07 4C 5E 04   B6
2851 : A9 01 85 88 20 00 FE A9   3F
2859 : 04 2C 00 18 D0 FB A2 28   3E
2861 : CA D0 FD 20 CE 81 A5 88   B7
2869 : F0 08 A9 FE 20 45 06 4C   84
2871 : C3 03 BA 86 86 C4 86 F0   C0
2879 : 05 A9 FF 20 45 06 2C A5   D7
2881 : 03 10 06 20 3B 06 20 6B   4D
2889 : 06 A5 82 20 45 06 A0 00   0E
2891 : A6 83 B5 8A F0 01 C8 CA   87
2899 : D0 F8 98 85 86 09 80 20   AF
28A1 : 45 06 A5 86 F0 0D A6 83   3D
28A9 : B5 8A F0 04 8A 20 45 06   2B
28B1 : CA D0 F5 4C A6 03 A0 0A   03
28B9 : A2 00 CA D0 FD 88 D0 FA   85
28C1 : 60 20 AE E9 AD 0F 18 29   20
28C9 : FB AE A4 03 E0 01 F0 02   83
28D1 : 09 04 8D 0F 18 AD 01 18   45
28D9 : 6A B0 09 20 13 06 AD 01   FC
28E1 : 18 6A 90 1C AD 00 1C 48   B2
28E9 : 29 FC 85 86 68 29 03 AA   F4
28F1 : CA 10 02 A2 03 8A 05 86   3E
28F9 : 8D 00 1C 20 B7 05 F0 DE   B7
2901 : AD 00 1C 29 03 D0 DD 20   49
2909 : B7 E9 A9 04 2C 00 18 D0   65
2911 : FB 60 A9 02 85 86 AD 00   2A
2919 : 1C 48 29 FC 85 87 68 29   CC
2921 : 03 AA E8 E0 04 D0 02 A2   E4
2929 : 00 8A 05 87 8D 00 1C 20   2A
2931 : B7 05 C6 86 D0 E0 60 A9   D6
2939 : 00 2C A9 01 A2 2A 95 8A   C1
2941 : CA D0 FB 60 8D 0C 40 A9   0C
2949 : 08 2C 0D 40 F0 FB 60 2C   7B
2951 : 00 1C 10 12 AD 0D 1C 0A   6D
2959 : 10 F5 AD 04 1C C6 85 D0   FF
2961 : EE 68 68 4C A7 07 2C 01   8C
2969 : 1C 60 A9 44 8D 06 1C 8D   3D
2971 : 04 1C A9 C5 8D 07 1C 8D   43
2979 : 05 1C AD 0B 1C 29 7F 8D   7D
2981 : 0B 1C 2C 04 1C A9 08 85   60
```

Listing 1. »Double-Touch«

```
2989 : 85 A0 00 84 89 A5 83 85  CE
2991 : 80 20 50 06 A2 09 2C 0F  37
2999 : 18 30 FB AD 01 1C C9 52  3B
29A1 : D0 EF 99 00 02 C8 2C 0F  04
29A9 : 18 30 FB AD 01 1C 99 00  E5
29B1 : 02 C8 CA D0 F1 20 50 06  51
29B9 : 84 87 A9 00 85 8A A9 40  3F
29C1 : 85 86 2C 0F 18 30 FB AE  C7
29C9 : 01 1C BC 0D A0 8A 29 07  BA
29D1 : 85 52 2C 0F 18 30 FB AD  BB
29D9 : 01 1C 85 53 29 C0 05 52  05
29E1 : AA 98 1D 0D 9F C9 07 F0  07
29E9 : 2E 4C 9C 07 2C 0F 18 30  41
29F1 : FB AE 01 1C BC 0D A0 8A  D3
29F9 : 29 07 85 52 2C 0F 18 30  4D
2A01 : FB AD 01 1C 85 53 29 C0  B0
2A09 : 05 52 AA 98 1D 0D 9F 45  38
2A11 : 8A 85 8A C6 86 F0 5F A6  94
2A19 : 53 BC 0D A1 8A 29 01 85  43
2A21 : 53 2C 0F 18 30 FB AD 01  ED
2A29 : 1C 85 54 29 F0 05 53 AA  1C
2A31 : 98 1D 0F 9F 45 8A 85 8A  E3
2A39 : A5 54 29 0F 85 54 2C 0F  FE
2A41 : 18 30 FB AD 01 1C A8 29  0C
2A49 : 80 05 54 AA BD 1D 9F 19  2C
2A51 : 0D A2 45 8A 85 8A 98 29  B3
2A59 : 03 85 55 2C 0F 18 30 FB  64
2A61 : AD 01 1C A8 29 E0 05 55  03
2A69 : AA BD 2A 9F 19 0D A3 45  83
2A71 : 8A 85 8A 4C ED 06 A6 53  3A
2A79 : BC 0D A1 8A 29 01 85 53  CD
2A81 : 2C 0F 18 30 FB AD 01 1C  AA
2A89 : 29 F0 05 53 AA 98 1D 0F  D8
2A91 : 9F C5 8A D0 06 A6 89 A9  DF
2A99 : 00 95 A0 A4 87 E6 89 C6  84
2AA1 : 80 F0 03 4C 92 06 A5 89  E7
2AA9 : F0 4B A9 02 85 31 A9 00  72
2AB1 : 85 30 85 34 85 86 20 D9  F7
2AB9 : 98 A4 86 A2 00 B5 52 99  C3
2AC1 : 00 02 C8 F0 09 E8 E0 04  76
2AC9 : D0 F3 84 86 F0 E8 C8 B9  72
2AD1 : 01 02 59 02 02 59 03 02  65
2AD9 : 59 04 02 D9 00 02 D0 0C  5C
2AE1 : 98 4A 4A 4A AA B5 A0 BE  D3
2AE9 : 01 02 95 8B 98 18 69 08  C2
2AF1 : A8 C6 89 D0 DA 60 00 00  29
2AF9 : 00 00 00 00 00 00 00 78  EA
2B01 : A9 00 8D 00 18 20 7C 87  91
2B09 : 20 64 87 AD 0F 18 09 20  09
2B11 : 8D 0F 18 AD 0C 1C 09 0E  C4
2B19 : 8D 0C 1C 20 CE 81 20 B2  96
2B21 : 81 2C 0D 40 A0 A0 A2 05  A7
2B29 : A9 08 2C 0D 40 F0 FB AD  5A
2B31 : 0C 40 99 00 03 C8 D0 F0  5F
2B39 : EE 35 03 CA D0 EA A9 04  EF
2B41 : 2C 00 18 F0 FB 2C 00 18  E3
2B49 : D0 FB A2 FE AD 5D 03 49  04
2B51 : A0 8D 5D 03 20 65 03 CA  3E
2B59 : D0 FA E8 E0 FE F0 ED 20  6C
2B61 : 65 03 D0 F6 8A 48 49 FF  6B
2B69 : A8 20 7C 87 A9 04 2C 00  9D
2B71 : 18 D0 13 CA D0 F6 20 88  66
2B79 : 87 A9 04 2C 00 18 D0 06  6C
2B81 : 88 D0 F6 68 AA 60 20 7C  63
2B89 : 87 A9 04 2C 00 18 D0 FB  67
2B91 : 2C 0D 40 4C A0 03 00 00  00
2B99 : 00 00 00 00 00 00 00 4C  32
2BA1 : A5 03 46 00 A9 02 8D 04  42
2BA9 : 40 A9 00 8D 05 40 A9 11  8B
2BB1 : 8D 0E 40 EE A4 03 CE A3  18
2BB9 : 03 10 0D A9 04 2C 00 18  0F
2BC1 : F0 FB 20 88 87 6C FC FF  98
2BC9 : AD A4 03 C9 01 F0 04 C9  FE
2BD1 : 24 D0 06 20 AC 04 4C E7  CF
2BD9 : 03 A9 04 2C 00 18 F0 FB  B4
2BE1 : 20 AE E9 20 F4 04 20 A1  0A
2BE9 : 04 20 CE 81 AD A4 03 C9  81
2BF1 : 24 30 03 38 E9 23 AA BD  D3
2BF9 : 2B 94 85 82 85 83 AD 00  4B
2C01 : 1C 29 9F 1D 08 94 8D 00  99
2C09 : 1C A9 44 8D 06 1C 8D 04  3C
2C11 : 1C A9 C5 8D 07 1C 8D 05  B6
2C19 : 1C AD 0B 1C 29 7F 8D 0B  2D
2C21 : 1C 2C 04 1C A9 08 85 85  D4
2C29 : 20 74 04 A2 09 A0 00 2C  C7
2C31 : 0F 18 30 FB AD 01 1C C9  BF
2C39 : 52 D0 ED 8D 0C 40 20 8F  83
2C41 : 04 20 74 04 A2 45 A0 01  CC
2C49 : 20 8F 04 C6 82 F0 03 4C  5F
2C51 : 29 04 A9 08 2C 0D 40 F0  F6
2C59 : F9 20 B7 E9 20 B2 81 A9  7E
2C61 : 08 2C 0D 40 F0 FB AD 0C  88
2C69 : 40 C9 FE D0 03 4C C3 03  0F
2C71 : 4C B4 03 2C 00 1C 10 12  A3
2C79 : AD 0D 1C 0A 10 F5 AD 04  65
2C81 : 1C C6 85 D0 EE 68 68 4C  E8
2C89 : 5A 04 2C 01 1C 60 2C 0F  A4
2C91 : 18 30 FB AD 01 1C 8D 0C  B5
2C99 : 40 CA D0 F2 88 10 EF 60  5A
2CA1 : A0 0A A2 00 CA D0 FD 88  2B
2CA9 : D0 FA 60 A9 04 2C 00 18  15
2CB1 : F0 FB 20 AE E9 AD 0F 18  F5
2CB9 : 29 FB AE A4 03 E0 01 F0  3D
2CC1 : 02 09 04 8D 0F 18 AD 01  65
2CC9 : 18 6A B0 09 20 F4 04 AD  79
2CD1 : 01 18 6A 90 06 20 1C 05  67
2CD9 : 4C D0 04 AD 00 1C 29 03  D0
2CE1 : D0 F3 20 A1 04 20 B7 E9  DB
2CE9 : A9 04 2C 00 18 D0 FB 20  D8
2CF1 : AE E9 60 A9 02 85 86 AD  A3
2CF9 : 00 1C 48 29 FC 85 87 68  29
2D01 : 29 03 AA E8 E0 04 D0 02  E9
2D09 : A2 00 8A 05 87 8D 00 1C  0C
2D11 : 20 A1 04 C6 86 D0 E0 20  8E
2D19 : A1 04 60 AD 00 1C 48 29  DE
2D21 : FC 85 86 68 29 03 AA CA  79
2D29 : 10 02 A2 03 8A 05 86 8D  49
2D31 : 00 1C 20 A1 04 60 FF 00  BF
```

Listing 1. »Double-Touch« wird mit dem MSE (Seite 159) im C64-Modus eingegeben

Benzinverbrauch im Griff

Es ist immer sinnvoll, den Überblick über Kraftstoffverbrauch und Kosten seines Autos zu behalten. Unser elektronisches Fahrtenbuch erstellt Monats- und Jahresübersichten und veranschaulicht die Ergebnisse durch übersichtliche Balkendiagramme.

Das Programm »Fahrtenbuch 80« ist gedacht für alle, die einen C 128 besitzen und ein komfortables und leistungsfähiges Hilfsmittel für Verbrauchsberechnungen ihres Kraftfahrzeuges suchen. »Fahrtenbuch 80« ist in Basic 7.0 geschrieben, wird über einfache Menüs bedient und benötigt einen Monitor mit 80-Zeichen-Darstellung. Übersichtliche Balkendiagramme zeigen Ihnen beispielsweise den durchschnittlichen Benzinverbrauch über das gesamte Jahr (Bild 1). Die Grafik kann auch als Hardcopy auf den Drucker ausgegeben werden.

Komfortable Bedienung

Geladen und gestartet wird das Programm mit »RUN "FAHRTENBUCH 80"«. Es meldet sich mit seinem Namen und »***«. Immer, wenn diese drei Sternchen auf dem Bildschirm erscheinen, erwartet das Programm das Betätigen der RETURN-Taste, um mit der Programmausführung fortzufahren. Nach <RETURN> gelangt man in das Hauptmenü. Die Anwahl der einzelnen Menüpunkte erfolgt durch die Eingabe der entsprechenden Zahl und <RETURN>. Mit <X> gelangt man wieder in das jeweils übergeordnete Menü bzw. zum Programmende. Die F1-Taste ist mit <X> <RETURN> belegt. Dies erleichert es, schnell aus den unteren Menüs zurück in das Hauptmenü zu kommen. Das Hauptmenü besteht aus drei Punkten, die noch weiter untergliedert sind: »Dateneingabe«, »Ausgabe«, »Diskettenoperationen«; außerdem aus dem Punkt »Programmende«.

Menüpunkt »Dateneingabe«

»Fahrtenbuch 80« verwaltet im Hauptspeicher jeweils Daten für ein komplettes Kalenderjahr. In jedem Monat können fünf Datenblöcke, bestehend aus Datum, Kilometerstand, getankten Litern Kraftstoff und Kosten für die Tankfüllung eingetragen werden. Das Programm errechnet daraus die gefahrenen Kilometer seit dem letzten Tanken, den entsprechenden Durchschnittsverbrauch pro 100 Kilometer sowie die Kosten pro Liter Benzin. Ferner wird pro Kalendermonat eine Übersicht erstellt, die alle Werte für den eingebenen Zeitraum enthält.

Untermenü »Neueingaben«

Nach Anwahl dieses Menüpunktes werden automatisch alle im Arbeitsspeicher befindlichen Daten gelöscht. Zunächst sind die Jahreszahl, die gleichzeitig als Dateiname dient, und die Startkilometer, d.h. der letzte Kilometerstand des Vorjahres (vollgetankt) oder <0> (z.B. bei Neukauf des Fahrzeuges), einzugeben. Daraufhin wird die erste Eingabe für Januar (siehe Datumsfeld) erwartet. Bei jedem Monat dürfen maximal fünf Datenblöcke eingegeben werden, was in der Regel ausreichend sein dürfte. Wird jedoch öfter getankt, so muß man die überzähligen Angaben vor der Eingabe zusammenfassen. Eine optimale Auswertung der Daten erfolgt, wenn immer vollgetankt wird.

Gibt man eine <0> als Datum ein oder drückt einfach <RETURN> (ohne Datumsangabe), so gelangt man in die Eingabemaske für den Folgemonat. Das gleiche geschieht, wenn bereits alle fünf Datenblöcke eines Monats eingegeben wurden.

Im Feld »km-Stand« ist der jeweilige Tachostand beim Tanken einzutragen, dann in die entsprechenden Felder die getankten Liter sowie die Kosten der Tankfüllung in DM. Dabei ist zu beachten, daß eventuelle Pfennigbeträge nach einem Dezimalpunkt, also nicht nach einem Komma, das vom Programm auch gar nicht angenommen wird, stehen müssen. Sind alle Angaben richtig, so bestätigt man dies bei der Sicherheitsabfrage mit <R>, und der Datenblock wird übernommen. Bei <F> wird der eingegebene Block vollständig gelöscht, und die Eingabe kann erneut erfolgen. Die Eingabemaske wird verlassen, wenn man in eines der Felder ein <X> einträgt und <RETURN> drückt, wobei eventuelle Einträge in anderen Datenfeldern ignoriert werden.

Untermenü »Ergänzungen«

Bei diesem Menüpunkt wird zunächst der erste mögliche Eingabemonat errechnet, zu dem noch Datenblöcke ergänzt werden können. Die entsprechende Eingabemaske erscheint, worauf wie bei »Neueingaben« Daten eingetragen werden dürfen. Gegebenenfalls muß man sich erst durch ein- oder mehrmaliges <X> bis zu dem gewünschten Monat nach vorne »blättern«.

Menüpunkt »Ausgabe (Bildschirm/Drucker)«

Nach getaner Arbeit werden hier die eingegebenen Daten und die daraus errechneten Werte übersichtlich dargestellt. Zuerst wählt man das Ausgabemedium, wobei »Drucker« oder »Bildschirm« zur Verfügung stehen. Die

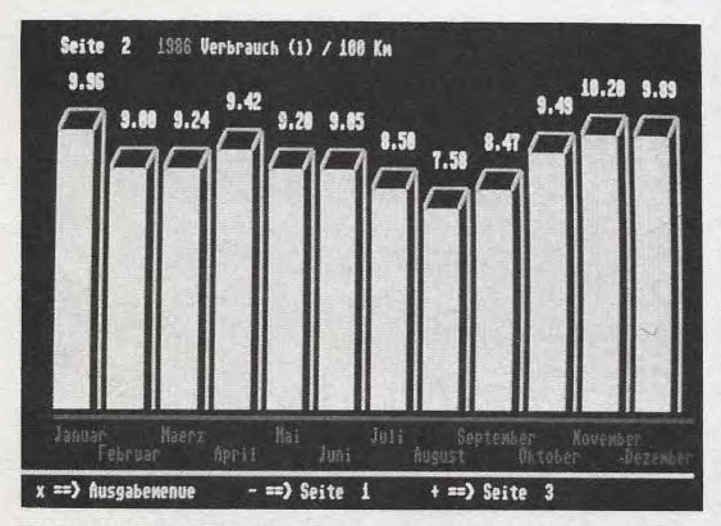


Bild 1. Der Benzinverbrauch unseres Beispielautos wird über ein Jahr übersichtlich in Diagrammform dargestellt

Ausgabe auf Diskette, also die Speicherung der Datenfiles, erfolgt im Menüpunkt »Diskettenoperationen«.

Untermenü »Ausgabeoptionen«

In Bild 2 sehen Sie das Menü mit den verschiedenen Ausgabemöglichkeiten. Mit dem ersten Unterpunkt wird die »Monatsübersicht« aufgerufen. Hier muß man zunächst den ersten Monat eingeben, der angezeigt werden soll; zulässig sind dabei Werte von »1« bis »12«. Weitere Monate lassen sich dann über die entsprechende Zahl in der Auswahlzeile aufrufen. Ist der Wert »0« oder größer als »12«, so wird der jeweilige Folgemonat dargestellt. Die Eingabe von <X> oder <F1> bewirkt einen Rücksprung ins Menü.

Einen Jahresüberblick nach verschiedenen Kriterien

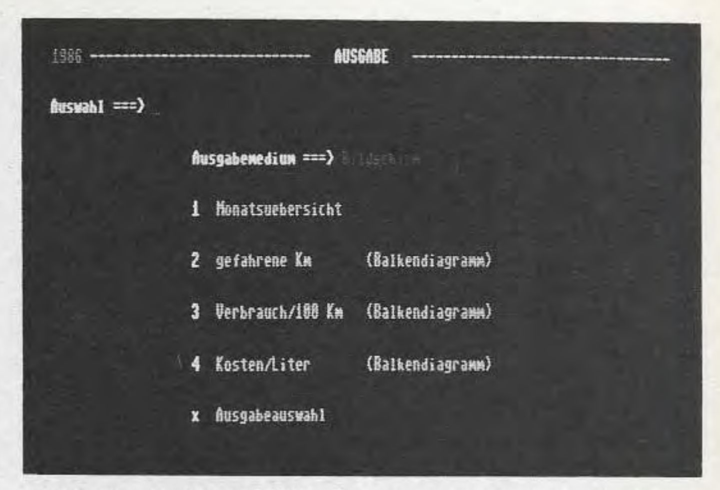


Bild 2. Das Untermenü für die Ausgabeoptionen auf den Bildschirm und Drucker

erhält man mit den drei anderen Menüpunkten. Dabei werden unter Verwendung der Monatsübersichten Balkendiagramme errechnet und angezeigt (Bild 1). Mit <+> oder <-> kann man das nächste bzw. vorhergehende Balkendiagramm aufrufen, durch <X> gelangt man wiederum in das übergeordnete Menü.

Menüpunkt »Diskettenoperationen«

In diesem Menübereich erfolgen Lade- und Speichervorgänge der Datensätze. Dabei dient das erste Untermenü, »Datenfiles anzeigen«, dazu, in Form eines eingeschränkten Directorys anzuzeigen, ob und zu welchem Jahr Datenfiles auf der im Laufwerk befindlichen Diskette abgelegt wurden.

Untermenü »Daten einlesen«

Wurden bereits Datenfiles gespeichert, so erfolgt durch Eingabe der gewünschten Jahreszahl das Einlesen dieser Angaben in den Arbeitsspeicher. Die dort eventuell befindlichen Daten werden überschrieben. Auf der Programmservice-Diskette sind unter File »1986« bereits Beispieldaten (wie Bild 1) vorhanden, die die einzelnen Ausgabemöglichkeiten (insbesondere die Balkendiagramme) demonstrieren sollen. Will man diese Daten einlesen, ist bei der Frage nach dem einzulesenden Jahr also »1986« einzugeben. In gleicher Weise geht man vor, wenn man später eigene Datenfiles gespeichert hat und diese laden und weiterverarbeiten will.

Untermenü »Daten speichern«

Das Sichern der unter »Neueingaben« oder »Ergänzungen« eingegebenen Werte auf Diskette erfolgt wahlweise unter der vorgegeben oder einer anderen (neu einzugebenden) Jahreszahl. Nach erfolgter Speicherung erhalten die im Arbeitsspeicher befindlichen Daten den gleichen Namen (Jahreszahl), unter dem sie eben auf Diskette geschrieben wurden.

Menüpunkt »Programmende«

Durch Anwahl dieses Menüpunktes wird das Programm verlassen. Mit »FAST:GOTO 30« erfolgt ein erneuter Programmstart ohne Datenverlust.

Fehlererkennung

Das Programm verfügt über eine eingeschränkte Fehlererkennung bei der Benutzung von Drucker und Diskettenstation. So wird bei Anwahl der Ausgabe auf dem Drucker zunächst dessen Ansprechbarkeit überprüft. Ist der Drucker nicht angeschlossen oder nicht eingeschaltet, so gelangt man nicht in das folgende Menü.

Ist der Drucker jedoch gerade aktiv, also druckend, und wird dann abgeschaltet, so kann dies nicht abgefangen werden, und das Programm steigt mit einem »DEVICE NOT PRESENT«-Error aus.

Bei Diskettenoperationen sieht es ähnlich aus. Auch hier wird nur am Anfang des Zugriffs das Vorhandensein der Floppystation abgefragt und der Fehlerkanal gelesen. Eine eventuelle Fehlermeldung erfolgt in englischer Sprache (Einzelheiten siehe Bedienungsanleitung des Laufwerks). In keinem Falle sollte die Diskette während eines Diskettenzugriffs aus dem Laufwerk entfernt oder das Laufwerk abgeschaltet werden.

Druckeranpassung

Das Programm ist in der vorliegenden Version auf den Drucker Commodore MPS 1000 (alle-DIP-Schalter auf »OFF«) abgestimmt. Die Druckausgabe läßt sich jedoch auch leicht an andere Drucker am seriellen Port mit Primäradresse 4 und Sekundäradresse 7 (für Groß-/Kleinschrift) anpassen.

Die einzigen Programmänderungen, die eventuell vorgenommen werden müssen, beziehen sich auf die Zeilen 2770, 2850, 3050 und 3270 des Listings. An diesen Stellen wird nämlich der Zeilenabstand eingestellt, damit die Balkendiagramme durchgehend gedruckt werden und zwischen den Zeilen der Monatsübersicht nicht zu große Lücken entstehen. Gegebenenfalls muß man ein wenig mit den Werten für die Veränderung des Zeilenabstandes des jeweiligen Druckes experimentieren, bis man ein optimales Ergebnis erhält.

Hier die entscheidenden Stellen im Listing:

```
2770 ..... :OPEN 6,4,6: PRINT#6,CHR$(22): ....
```

Öffnen des Befehlskanals zum Drucker, Einstellen des Zeilenabstandes.

```
2850 ..... :PRINT#6,CHR$(36): CLOSE 6: ....
```

Einstellen des Grundwertes für den Zeilenabstand. Gleiches gilt für die Zeilen

```
3050 ..... :OPEN 6,4,6: PRINT#6,CHR$(24): ....und
3270 ..... :PRINT#6,CHR$(36): CLOSE 6: ...
```

Mit »Fahrtenbuch 80« haben Sie ein sehr nützliches Werkzeug an der Hand, um jederzeit über die Kosten Ihres Autos informiert zu sein. Es wird sich dann vielleicht für manchen herausstellen, daß der »Super-Kauf«, den er meinte gemacht zu haben, sich als wahres »Groschengrab« entpuppt. Ein weiterer positiver Aspekt kann sein, daß durch die ständige Präsenz der Kosten so mancher seine Fahrweise ändert und sowohl Geldbeutel als auch Umwelt entlastet.

(Wolfgang Sartor/Dr. Rudolf Egg)

```
10 GRAPHIC 5: CLR : COLOR 6,1: PRINT CHR$(147);CH
   R$(14);CHR$(11);CHR$(27)+"G": KEY 1,"X"+CHR$(1
   3)                                              <PQ3>
20 GOSUB 1020: FAST                                <GVU>
30 PRINT "{CLR,PURPLE}"J$"{CYAN} ----------------
   ------------{3SPACE}HAUPTMENUE{3SPACE}---------
   -----------------------"                        <E8V>
40 IF F=1 THEN PRINT "{HOME}";TAB(63);F$: F=0      <3I8>
50 PRINT AU$+CHR$(27)+"Q"                          <UIM>
60 PRINT D$;"1{GREEN,2SPACE}DATENEINGABE"          <UIM>
70 PRINT D$;"2{GREEN,2SPACE}AUSGABE (BILDSCHIRM /
   DRUCKER)"                                       <30H>
80 PRINT D$;"3{GREEN,2SPACE}DISKETTENOPERATIONEN"  <1BJ>
90 PRINT D$;"X{GREEN,2SPACE}PROGRAMMENDE"          <DFI>
100 Z=1: PU=1: PRINT "{HOME,3DOWN}" TAB(13)"{RED}
    @";: GOSUB 1290: A=ASC(B$)                     <SRV>
110 IF A=88 THEN PRINT "{CLR,CYAN}";CHR$(12): SLO
    W : END                                        <28N>
120 IF A<49 OR A>51 THEN F=1: GOTO 40              <2J8>
130 A=A-48                                         <M6E>
140 ON A GOSUB 150,750,870: GOTO 30                <D0K>
150 PRINT "{CLR,PURPLE}"J$"{CYAN} ---------------
    ------------{3SPACE}DATENEINGABE{3SPACE}------
    -----------------------"                       <LOI>
160 IF F=1 THEN PRINT "{HOME}";TAB(63);F$: F=0     <RH8>
170 PRINT AU$+CHR$(27)+"Q"                         <TIM>
180 PRINT D$;"1{GREEN,2SPACE}NEUEINGABEN"          <LTO>
190 PRINT D$;"2{GREEN,2SPACE}ERGAENZUNGEN"         <QCJ>
200 PRINT HM$                                      <QHV>
210 Z=1: PU=1: PRINT "{HOME,3DOWN}" TAB(13)"{RED}
    @";: GOSUB 1290: A=ASC(B$)                     <0OF>
220 IF A=88 THEN RETURN                            <K6M>
230 IF A<49 OR A>50 THEN F=1: GOTO 160             <U25>
240 A=A-48                                         <64E>
250 ON A GOSUB 260,350: GOTO 150                   <L14>
260 REM *** EINGABEN NEU ***                       <IIA>
270 PRINT "{HOME,17DOWN}" TAB(20)"{RED,CTRL+O}DAT
    EN IM HAUPTSPEICHER WERDEN GELOESCHT !"+CHR$(
    27)+"O"                                        <FR3>
280 FOR I=1 TO 60: FOR J=1 TO 7: S(I,J)=0: NEXT J
    : NEXT I                                       <PGU>
290 PRINT "{HOME,17DOWN,GREEN}JAHRESZAHL{2SPACE,C
    YAN}===> {RED}@"+CHR$(27)+"Q";: Z=4: PU=1: GO
    SUB 1290: PRINT " ": IF F=2 THEN RETURN : ELS
    E BEGIN                                        <VMQ>
300 J$=B$: I=LEN(J$)                               <VI8>
310 IF I<4 THEN J$=J$+LEFT$(SP$,4-I)               <P7E>
320 BEND                                           <APE>
330 PRINT "{2DOWN,GREEN}START KM{4SPACE,CYAN}===>
     {RED}@";: Z=10: PU=0: GOSUB 1290: IF F=2 THE
    N J$="9999": RETURN : ELSE KM=INT(B*C)/C       <8LR>
340 MO=1: T=1: GOSUB 460: RETURN                   <SQR>
350 REM *** ERGAENZUNGEN ***                       <IQB>
360 IF S(60,1)<>0 THEN PRINT "{HOME,17DOWN}" TAB(
    20)"{RED,CTRL+O}FUER {CYAN}";J$;"{RED} IST KE
    INE EINGABE MEHR MOEGLICH !"+CHR$(27)+"O": SL
    EEP 5: RETURN                                  <7J4>
370 T=61                                           <48K>
380 DO                                             <UNR>
390 T=T-1                                          <8T9>
400 LOOP UNTIL (S(T,1)<>0 OR T=0)                  <20P>
410 IF T=0 THEN GOSUB 260: RETURN : REM *** NEUEI
    NGABEN ***                                     <E7V>
420 KM=S(T,2)                                      <3HU>
430 MO=INT(T/5)+1: T=T+1-INT((T+1)/5)*5: IF T=0 T
    HEN T=5                                        <954>
440 GOSUB 460: RETURN                              <47A>
450 REM *** EINGABEN LESEN ***                     <HA5>
460 DO                                             <UNR>
470 DO                                             <VNV>
480 K=(MO-1)*5+T                                   <CNU>
490 GOSUB 1570                                     <JJ1>
500 PRINT "{HOME,3DOWN}" TAB(24);S1$;"{RED}@";: Z
    =2: PU=1: GOSUB 1290: PRINT "{LEFT} ": IF F=2
     THEN EXIT                                     <B0D>
510 GOSUB 1670: IF F=1 THEN F=0: GOTO 500          <74O>
520 IF F=3 THEN F=0: EXIT                          <DK9>
530 PRINT "{HOME,7DOWN}" TAB(35);S2$;"{RED}@";: Z
    =10: PU=0: GOSUB 1290: PRINT "{LEFT} ": IF F=
    2 THEN EXIT                                    <J7J>
540 GOSUB 1730: IF F=1 THEN F=0: GOTO 530          <7IQ>
550 PRINT "{HOME,9DOWN}" TAB(35);S2$;"{RED}@";: Z
    =10: PU=0: GOSUB 1290: PRINT "{LEFT} ": IF F=
    2 THEN EXIT                                    <VJ1>
560 GOSUB 1770: IF F=1 THEN F=0: GOTO 550          <62S>
570 PRINT "{HOME,11DOWN}" TAB(35);S2$;"{RED}@";:
    Z=10: PU=0: GOSUB 1290: PRINT "{LEFT} ": IF F
    =2 THEN EXIT                                   <DPA>
580 GOSUB 1810: IF F=1 THEN F=0: GOTO 570          <6MT>
590 GOSUB 1920: IF F=1 THEN F=0: GOTO 490          <TNH>
600 GOSUB 1850                                     <J3F>
610 LOOP UNTIL T=6                                 <C5J>
620 T=1: MO=MO+1                                   <793>
630 LOOP UNTIL (MO=13 OR F=2)                      <NI3>
640 IF F=2 THEN FOR I=1 TO 4: S(K,I)=0: NEXT I: F
    =0                                             <6I3>
650 REM *** MONATSUEBERSICHT BERECHNEN ***         <GA1>
660 FOR I=1 TO 12                                  <Q4V>
670 FOR L=1 TO 3: R(I,L)=0: NEXT L                 <70D>
680 FOR J=1 TO 5: K=(I-1)*5+J                      <MOB>
690 IF S(K,1)<>0 THEN BEGIN                        <60E>
700 R(I,1)=R(I,1)+S(K,5): R(I,2)=R(I,2)+S(K,3): R
    (I,3)=R(I,3)+S(K,4)                            <11G>
710 BEND : ELSE IF J=1 THEN GOTO 740               <NDS>
720 NEXT J                                         <BH5>
730 R(I,3)=INT(R(I,3)/R(I,2)*C*10)/(C*10): R(I,2)
    =INT(R(I,2)/R(I,1)*C*C)/C                      <B9I>
740 NEXT I: RETURN                                 <S82>
750 PRINT "{CLR,PURPLE}"J$"{CYAN} ---------------
    ------------{3SPACE}AUSGABEAUSWAHL{2SPACE}---
    -----------------------------"                 <U1L>
760 IF F=1 THEN PRINT "{HOME}";TAB(53);"{CYAN}---
    --------";F$: F=0                              <FN8>
770 IF F=3 THEN F=0: PRINT "{HOME}";TAB(53);"{CTR
    L+O,RED} DRUCKER NICHT ANSPRECHBAR"+CHR$(27)+
    "O"                                            <5EG>
780 PRINT AU$+CHR$(27)+"Q"                         <5HM>
790 PRINT D$;"1{GREEN,2SPACE}DRUCKAUSGABE"         <PG7>
800 PRINT D$;"2{GREEN,2SPACE}BILDSCHIRMAUSGABE"    <8NN>
810 PRINT HM$                                      <VPV>
820 Z=1: PU=1: PRINT "{HOME,3DOWN}" TAB(13)"{RED}
    @";: GOSUB 1290: A=ASC(B$)                     <SJV>
830 IF A=88 THEN RETURN                            <HE3>
840 IF A<49 OR A>50 THEN F=1: GOTO 760             <R69>
```

```
850 D=A-48: IF D=1 THEN CLOSE 3: OPEN 3,4,7: CLOS
    E 3: IF ST<0 THEN F=3: GOTO 770                  <6P3>
860 GOSUB 2310: GOTO 750                             <A2E>
870 PRINT "{CLR,PURPLE}"J$"{CYAN} ---------------
    --------{3SPACE}DISKETTENOPERATIONEN{3SPACE}--
    -----------------------"                         <DR5>
880 IF F=1 THEN PRINT "{HOME}";TAB(52)"{CYAN}-----
    --------";F$: F=0                                <GCK>
890 IF F=3 THEN F=0: PRINT "{HOME}";TAB(52);"{CTR
    L+O,RED} FLOPPY 8 NICHT ANSPRECHBAR"+CHR$(27)
    +"O"                                             <U4E>
900 PRINT AU$+CHR$(27)+"Q"                           <55M>
910 PRINT D$;"1{GREEN,2SPACE}GESPEICHERTE DATENFI
    LES ANZEIGEN"                                    <4A7>
920 PRINT D$;"{2UP,3SPACE,GREEN}(EINGESCHRAENKTES
     DIRECTORY)"                                     <E06>
930 PRINT D$;"2{GREEN,2SPACE}DATEN EINLESEN"         <UGO>
940 PRINT D$;"3{GREEN,2SPACE}DATEN ABSPEICHERN"      <813>
950 PRINT HM$                                        <U9V>
960 Z=1: PU=1: PRINT "{HOME,3DOWN}" TAB(13)"{RED}
    █";: GOSUB 1290: A=ASC(B$)                       <4GV>
970 IF A=88 THEN RETURN                              <GU5>
980 IF A<49 OR A>51 THEN F=1: GOTO 880               <J6F>
990 CLOSE 3: OPEN 3,8,3: CLOSE 3: IF ST<0 THEN F=
    3: GOTO 890                                      <APS>
1000 A=A-48                                          <ACE>
1010 ON A GOSUB 2000,2050,2150: GOTO 870             <FUA>
1020 REM *** INITIALISIERUNG ***                     <3KE>
1030 FAST : DIM S(60,7),R(12,3),M(12),M$(12),B(3,
     12),B$(3): J$="9999": F=0: C=100                <GMH>
1040 B$(1)="#####": B$(2)="##.##": B$(3)="#.###":
      D$="{2DOWN,18SPACE,CYAN}"                      <P4I>
1050 HM$=D$+"X{GREEN,2SPACE}HAUPTMENUE"              <IS1>
1060 F$="{CTRL+O,RED} FALSCHE AUSWAHL"+CHR$(27)+"
     O"                                              <0D3>
1070 AU$="{2DOWN,CYAN}AUSWAHL ===> "                 <3P1>
1080 SP$="{30SPACE}"                                 <7KN>
1090 L$="{HOME,16DOWN}"                              <DGI>
1100 S1$="{2SPACE,2LEFT}"                            <IPK>
1110 S2$="{10SPACE,10LEFT}"                          <JN6>
1120 RESTORE                                         <VDM>
1130 FOR I=1 TO 12: READ M(I): NEXT I                <15B>
1140 FOR I=1 TO 12: READ M$(I): NEXT I               <6BR>
1150 DATA 31,29,31,30,31,30,31,31,30,31,30,31        <VVK>
1160 DATA "JANUAR","FEBRUAR","MAERZ","APRIL","MAI
     ","JUNI","JULI","AUGUST","SEPTEMBER","OKTOBE
     R","NOVEMBER","DEZEMBER"                        <LGD>
1170 SLOW : PRINT "{CLR,3DOWN,CYAN}" SPC(20): X$=
     "FAHRTENBUCH-PROGRAMM": GOSUB 1220              <MS3>
1180 PRINT "{2DOWN,CYAN}" SPC(20): X$="(W) WOLLI
     SOFTWARE{3SPACE}1986": GOSUB 1220               <191>
1190 PRINT "{2DOWN,GREEN}" SPC(20): X$="***█": GO
     SUB 1220: S$="{GREEN,2DOWN}"+X$                 <EAH>
1200 GOSUB 2550: RETURN                              <9QH>
1210 REM *** TITEL SOUND ***                         <2C9>
1220 FOR I=1 TO LEN(X$)                              <SCA>
1230 XT$=MID$(X$,I,1)                                <DVC>
1240 IF XT$<>" " THEN PRINT "{CTRL+G}";              <S7F>
1250 PRINT XT$;                                      <E4G>
1260 FOR T=1 TO 50: NEXT T                           <LAR>
1270 NEXT I: PRINT                                   <APQ>
1280 RETURN                                          <8C7>
1290 REM *** ZAHLENEINGABE LESEN ***                 <0C1>
1300 Y=0: B$="": F=0                                 <I62>
1310 DO                                              <JEF>
1320 GET KEY A$                                      <6GD>
1330 A=ASC(A$)                                       <QAR>
1340 IF A=13 THEN BEGIN                              <DJC>
1350 IF Z=2 THEN B=VAL(B$): PRINT "{LEFT} ";: PRI
     NT L$;SP$;SP$;: EXIT                            <KSD>
1360 IF B$="" THEN GOTO 1320: ELSE B=VAL(B$): PRI
     NT "{LEFT} ";: PRINT L$;SP$;SP$;: EXIT          <T07>
1370 BEND                                            <DPU>
1380 IF A=20 AND Y>0 THEN BEGIN                      <VCH>
1390 IF RIGHT$(B$,1)=CHR$(46) THEN PU=0              <U60>
1400 PRINT "{2LEFT}█ {LEFT}";: Y=Y-1: B$=LEFT$(B$
     ,Y): GOTO 1320                                  <2PN>
1410 BEND                                            <DTU>
1420 IF A=46 THEN BEGIN                              <FB8>
1430 IF PU=0 THEN PU=1: GOTO 1530                    <P3K>
1440 GOTO 1320                                       <30G>
1450 BEND                                            <TVU>
1460 IF A=88 AND Y=0 THEN BEGIN                      <V4M>
1470 Y=1: B$="X": PRINT "{LEFT}X█";                  <788>
1480 GET KEY A$: A=ASC(A$): IF A<>13 AND A<>20 TH
     EN GOTO 1480                                    <OJ2>
1490 IF A=13 THEN F=2: PRINT "{LEFT} ";: PRINT L$
     ;SP$;SP$;: EXIT                                 <90U>
1500 GOTO 1380                                       <0OQ>
1510 BEND                                            <5SU>
1520 IF A>47 AND A<58 THEN BEGIN                     <DUC>
1530 Y=Y+1: IF Y>Z THEN Y=Z: ELSE PRINT "{LEFT}";
     A$;"█";: B$=B$+A$                               <UVG>
1540 BEND                                            <HUE>
1550 LOOP                                            <S7P>
1560 RETURN                                          <8C7>
1570 REM *** EINGABEBILD ***                         <0C1>
1580 PRINT "{CLR,CYAN}---------------------------
     --------{3SPACE}EINGABE{3SPACE}--------------
     --------------------"                           <2DH>
1590 PRINT SPC(11)"{2DOWN,GREEN}DATUM{3SPACE,CYAN
     }===>{6SPACE,GREEN}[{CYAN}";M$(MO);" ";J$;"{
     GREEN}]"                                        <EMJ>
1600 PRINT SPC(11)"{3DOWN,GREEN}KM-STAND{11SPACE,
     CYAN}===> "                                     <SM8>
1610 PRINT SPC(11)"{DOWN,GREEN}GETANKTE LITER{5SP
     ACE,CYAN}===> "                                 <5H8>
1620 PRINT SPC(11)"{DOWN,GREEN}KOSTEN{2SPACE}(DM)
     {7SPACE,CYAN}===> "                             <FND>
1630 PRINT SPC(11)"{GREEN}(TANKFUELLUNG)             <QQ3>
1640 PRINT "{8DOWN}";: FOR I=1 TO 78: PRINT "█";:
      NEXT : PRINT                                   <OI3>
1650 PRINT "{3RIGHT,DOWN}X {CYAN}==>{GREEN} ENDE
     (DER EINGABE)"                                  <G6K>
1660 RETURN                                          <807>
1670 REM *** DATUM ZUWEISEN ***                      <1S7>
1680 B=INT(B): S(K,1)=B                              <3LB>
1690 IF B>M(MO) THEN PRINT L$;TAB(20)"{CTRL+O,RED
     }DATUM ZU GROSS"+CHR$(27)+"O";: F=1             <1MO>
1700 IF T>1 AND B<S(K-1,1) THEN PRINT L$;TAB(20)"
     {CTRL+O,RED}DATUM ZU KLEIN"+CHR$(27)+"O";: F
     =1                                              <U4Q>
1710 IF B=0 THEN F=3                                 <T1H>
1720 RETURN                                          <8C7>
1730 REM *** KM-STAND ZUWEISEN ***                   <0C1>
1740 B=INT(B*10)/10: S(K,2)=B                        <7EP>
1750 IF B<=KM THEN PRINT L$;TAB(20)"{CTRL+O,RED}K
     M-STAND ZU KLEIN"+CHR$(27)+"O";: F=1            <H3N>
1760 RETURN                                          <847>
1770 REM *** LITER ZUWEISEN ***                      <1C5>
1780 B=INT(B*C)/C: S(K,3)=B                          <6E2>
1790 IF B=0 THEN PRINT L$;TAB(20)"{CTRL+O,RED}GET
     ANKTE LITER ZU KLEIN"+CHR$(27)+"O";: F=1        <0F9>
1800 RETURN                                          <9K7>
1810 REM *** DM ZUWEISEN ***                         <7CT>
1820 B=INT(B*C)/C: S(K,4)=B                          <1QU>
1830 IF B=0 THEN PRINT L$;TAB(20)"{CTRL+O,RED}KOS
     TEN (DM) ZU KLEIN"+CHR$(27)+"O";: F=1           <AVC>
1840 RETURN                                          <9S7>
1850 REM *** BERECHNUNG ***                          <6CP>
1860 T=T+1                                           <7GF>
1870 S(K,5)=INT((S(K,2)-KM)*10)/10: REM *** GEF.
     KM ***                                          <0HQ>
1880 KM=S(K,2)                                       <NU3>
1890 S(K,6)=INT(S(K,3)*C*C/S(K,5))/C: REM *** VER
     BRAUCH / 100 KM ***                             <5VD>
1900 S(K,7)=INT(S(K,4)/S(K,3)*C*10)/(C*10): REM *
     ** DM / LITER ***                               <35N>
1910 RETURN                                          <PI7>
1920 REM *** EINGABEBESTAETIGUNG ***                 <74S>
1930 PRINT L$;TAB(11);"{GREEN}R {CYAN}==> {GREEN}
     EINGABE RICHTIG{6SPACE}F {CYAN}==>{GREEN} EI
     NGABE WIEDERHOLEN": F=0                         <529>
1940 DO                                              <2CB>
1950 GET KEY A$                                      <28D>
1960 IF A$="R" THEN F=0: EXIT                        <JE2>
1970 IF A$="F" THEN F=1: EXIT                        <363>
1980 LOOP                                            <8D9>
1990 RETURN                                          <P27>
2000 REM *** DIRECTORY ***                           <040>
2010 CLOSE 15: OPEN 15,8,15,"I": GOSUB 2500: CLOS
     E 15: IF F=1 THEN F=0: RETURN                   <0MC>
2020 WINDOW 49,10,79,24,1: PRINT "{CYAN}";: DIREC
     TORY "DATEN*"                                   <6M6>
2030 GOSUB 2500: GOSUB 2530                          <BC6>
2040 F=0: WINDOW 0,0,79,24: RETURN                   <K32>
2050 REM *** DATEN EINLESEN ***                      <1C5>
2060 PRINT "{CLR,CYAN}---------------------------
     ---{3SPACE}DATEN EINLESEN{3SPACE}-----------
     -------------------"                            <MQ8>
2070 PRINT D$"{4LEFT,GREEN}VON WELCHEM JAHR SOLLE
     N DIE DATEN EINGELESEN WERDEN ?": PRINT AU$;
     S2$;"{RED}█";: Z=4: PU=1: GOSUB 1290: IF F=2
      THEN RETURN                                    <TUL>
2080 J2$=B$: I=LEN(L$): IF I<4 THEN J2$=J2$+LEFT$
     (SP$,4-I)                                       <762>
2090 CLOSE 15: OPEN 15,8,15,"I": GOSUB 2500: CLOS
     E 15: IF F=1 THEN F=0: GOTO 2060                <2DL>
2100 DCLOSE #2: DOPEN #2,"DATEN"+J2$: GOSUB 2500:
      IF F=1 THEN F=0: DCLOSE #2: GOTO 2060          <AMS>
2110 PRINT "{HOME,17DOWN}" TAB(20)"{RED,CTRL+O}DA
     TEN FUER {CYAN}";J2$;"{RED} WERDEN GELESEN !
     "+CHR$(27)+"O"                                  <103>
2120 FOR I=1 TO 60: FOR J=1 TO 7: INPUT#2,S(I,J):
      NEXT J: NEXT I                                 <HSP>
2130 FOR I=1 TO 12: FOR J=1 TO 3: INPUT#2,R(I,J):
      NEXT J: NEXT I                                 <5MO>
2140 DCLOSE #2: J$=J2$: RETURN                       <A6F>
2150 REM *** DATEN ABSPEICHERN ***                   <1S7>
2160 PRINT "{CLR,CYAN}---------------------------
     -{3SPACE}DATEN ABSPEICHEICHERN{3SPACE}------
     -----------------"                              <CFF>
2170 PRINT D$"{GREEN,4LEFT}SOLLEN DIE DATEN UNTER
      {CYAN}"J$" {GREEN}ABGELEGT WERDEN{2SPACE}(J
     /N){2SPACE}?"                                   <95F>
2180 DO                                              <IEB>
2190 GET KEY A$: A=ASC(A$)                           <908>
2200 LOOP UNTIL (A=74 OR A=78)                       <FAP>
2210 IF A=78 THEN BEGIN                              <D79>
2220 PRINT D$;"{CYAN}JAHRESZAHL ===> ";S2$;"{RED}
```

Listing 1. »Fahrtenbuch 80« verschafft Ihnen Überblick zu Benzinverbrauch und Kosten. Bitte mit dem Checksummer 128 (Seite 158) eingeben.

```
   @";: Z=4: PU=1: GOSUB 1290: IF F=2 THEN RETU
   RN                                                   <6L3>
2230 J2$=B$: I=LEN(L$): IF I<4 THEN J2$=J2$+LEFT$
   (SP$,4-I)                                            <4UD>
2240 BEND : ELSE J2$=J$                                 <KTD>
2250 CLOSE 15: OPEN 15,8,15,"I": GOSUB 2500: CLOS
   E 15: IF F=1 THEN F=0: GOTO 2160                     <3TJ>
2260 DCLOSE #2: DOPEN #2,"@DATEN"+J2$,W: GOSUB 25
   00: IF F=1 THEN F=0: DCLOSE #2: GOTO 2160            <NC8>
2270 PRINT "{HOME,17DOWN}" TAB(20)"{RED,CTRL+O}DA
   TEN WERDEN UNTER {CYAN}";J2$;"{RED} ABGELEGT
    !"+CHR$(27)+"O"                                     <BCK>
2280 FOR I=1 TO 60: FOR J=1 TO 7: PRINT#2,S(I,J);
   CHR$(13);: NEXT J: NEXT I                            <O34>
2290 FOR I=1 TO 12: FOR J=1 TO 3: PRINT#2,R(I,J);
   CHR$(13);: NEXT J: NEXT I                            <997>
2300 DCLOSE #2: J$=J2$: RETURN                          <A6F>
2310 REM *** AUSGABE ***                                <1S7>
2320 PRINT "{CLR,PURPLE}"J$"{CYAN} ---------------
   ----------------{3SPACE}AUSGABE{3SPACE}--------
   ------------------------------"                      <Z3T>
2330 IF F=1 THEN PRINT "{HOME}";TAB(63);F$: F=0         <9V1>
2340 PRINT AU$+CHR$(27)+"Q"                             <3TM>
2350 PRINT D$;"AUSGABEMEDIUM ===> {RED}";: IF D=1
    THEN PRINT "DRUCKER": ELSE PRINT "BILDSCHIR
   M"                                                   <Q3I>
2360 PRINT D$;"1{GREEN,2SPACE}MONATSUEBERSICHT"         <K5C>
2370 PRINT D$;"2{GREEN,2SPACE}GEFAHRENE KM{7SPACE
   }(BALKENDIAGRAMM)"                                   <J2A>
2380 PRINT D$;"3{GREEN,2SPACE}VERBRAUCH/100 KM{3S
   PACE}(BALKENDIAGRAMM)"                               <SSO>
2390 PRINT D$;"4{GREEN,2SPACE}KOSTEN/LITER{7SPACE
   }(BALKENDIAGRAMM)"                                   <GVF>
2400 PRINT D$;"X{GREEN,2SPACE}AUSGABEAUSWAHL"           <3GG>
2410 Z=1: PU=1: PRINT "{HOME,3DOWN}" TAB(13)"{RED
   }@";: GOSUB 1290: A=ASC(B$)                          <JQF>
2420 IF A=88 THEN RETURN                                <50I>
2430 IF A<49 OR A>52 THEN F=1: GOTO 2330                <84R>
2440 W=A-48: CLOSE 3: OPEN 3,4,7: CLOSE 3: IF ST<
   0 AND D=1 THEN F=3: RETURN                           <5EI>
2450 IF W=1 THEN BEGIN                                  <G90>
2460 PRINT "{CLR,GREEN}START MONAT {CYAN}===> {RE
   D}@";: Z=2: PU=1: GOSUB 1290: IF F=2 THEN B=
   -1: F=0                                              <DHQ>
2470 IF B>0 AND B<13 THEN MO=B: GOSUB 3020: ELSE
   IF B<>-1 THEN F=1                                    <D07>
2480 BEND : ELSE W=W-1: GOSUB 2590                      <68U>
2490 GOTO 2320                                          <0GS>
2500 REM *** DISCFEHLER ? ***                           <2KA>
2510 IF DS<>0 THEN WINDOW 49,17,79,24,1: PRINT "{
   RED,CTRL+O,5SPACE}DISK - FEHLER"+CHR$(27)+"O
   ": PRINT "{CYAN,2DOWN}"DS$: F=1: GOSUB 2530:
    WINDOW 0,0,79,24                                    <94J>
2520 RETURN                                             <8S7>
2530 REM *** STERNCHEN ***                              <2C9>
2540 PRINT S$                                           <SLP>
2550 DO                                                 <NEV>
2560 GET KEY A$: A=ASC(A$)                              <E0K>
2570 LOOP UNTIL A=13                                    <CLK>
2580 RETURN                                             <887>
2590 REM *** BALKENDIAGRAMME ***                        <0S3>
2600 IF D=1 THEN P$="TTTT": V=34: ELSE P$="{RVSON
   ,4SPACE,RVOFF}": V=16                                <C9L>
2610 PRINT "{CLR}"D$"{CTRL+O,RED}BALKENDIAGRAMME
   WERDEN BERECHNET !"+CHR$(27)+"O": FOR I=1 TO
    3: B(I,0)=0.001: NEXT I                             <MHI>
2620 FOR I=1 TO 3: FOR J=1 TO 12                        <TN0>
2630 B(I,J)=R(J,I): IF B(I,J)>B(I,0) THEN B(I,0)=
   B(I,J)                                               <10Q>
2640 NEXT J: NEXT I                                     <QA0>
2650 FOR I=1 TO 3: FOR J=1 TO 12                        <1KG>
2660 B(I,J)=INT((B(I,J)/B(I,0))*V+.5): IF B(I,J)=
   1 THEN B(I,J)=2                                      <GPT>
2670 IF B(I,J)=0 THEN B(I,J)=-2                         <S1S>
2680 NEXT J: NEXT I                                     <2A0>
2690 DO                                                 <TFN>
2700 IF D=1 THEN BEGIN                                  <J40>
2710 PRINT "{CLR}" SPC(6)"{CYAN}SEITE ";W;" {PURP
   LE}";J$;"{GREEN} ";                                  <2R5>
2720 IF W=1 THEN PRINT "GEFAHRENE KM / MONAT": EL
   SE IF W=2 THEN PRINT "VERBRAUCH (L) / 100 KM
   ": ELSE PRINT "KOSTEN (DM) / LITER"                  <SVA>
2730 CLOSE 1: OPEN 1,4,7: CMD 1: FOR I=1 TO 79: P
   RINT "-";: NEXT : PRINT                              <ER6>
2740 BEND                                               <HQE>
2750 PRINT "{CLR}" SPC(6)"{CYAN}SEITE ";W;"{2SPAC
   E,PURPLE}";J$;"{GREEN} ";                            <J11>
2760 IF W=1 THEN PRINT "GEFAHRENE KM / MONAT": EL
   SE IF W=2 THEN PRINT "VERBRAUCH (L) / 100 KM
   ": ELSE PRINT "KOSTEN (DM) / LITER"                  <SNB>
2770 PRINT : PRINT CHR$(142);: IF D=1 THEN CLOSE
   1: CLOSE 6: OPEN 1,4: OPEN 6,4,6: PRINT#6,CH
   R$(22): CMD 1                                        <QNV>
2780 FOR I=V+2 TO 1 STEP -1                             <FU3>
2790 PRINT SPC(6);: FOR K=1 TO 12                       <2MA>
2800 IF B(W,K)=I-2 THEN PRINT "{CYAN}";: PRINT US
   ING B$(W);R(K,W);: PRINT " ";                        <KIQ>
2810 IF B(W,K)=I-1 THEN PRINT "{YELLOW} @@@@ ";         <4I9>
2820 IF B(W,K)=I   THEN PRINT "{YELLOW}N{3SPACE}M
   @";                                                  <29T>
2830 IF B(W,K)>=I+1 THEN PRINT "{GREEN}";P$;"{YEL
   LOW} @";                                             <PP6>
2840 IF B(W,K)<I-2 THEN PRINT "{6SPACE}";               <HKB>
2850 NEXT K: PRINT : NEXT I: IF D=1 THEN PRINT#6,
   CHR$(36): CLOSE 6: CLOSE 1: OPEN 1,4,7: CMD
   1                                                    <N0M>
2860 PRINT SPC(6);: FOR I=1 TO 73: PRINT "{LIG.BL
   UE}*";: NEXT I: IF D=1 THEN PRINT SPC(1): EL
   SE PRINT CHR$(14)                                    <B56>
2870 PRINT SPC(6)M$(1) SPC(6)M$(3) SPC(8)M$(5) SP
   C(8)M$(7) SPC(6)M$(9) SPC(4)M$(11)                   <TSB>
2880 PRINT SPC(11)M$(2) SPC(6)M$(4) SPC(7)M$(6) S
   PC(7)M$(8) SPC(6)M$(10) SPC(5)M$(12)                 <P71>
2890 IF D=1 THEN FOR I=1 TO 79: PRINT "-";: NEXT
   : PRINT#1: CLOSE 1: PRINT CHR$(14)                   <9R9>
2900 FOR I=1 TO 79: PRINT "{GREEN}*";: NEXT I: PR
   INT : VS=W-1-INT((W-1)/3)*3: IF VS=0 THEN VS
   =3                                                   <HKA>
2910 NS=W+1-INT((W+1)/3)*3: IF NS=0 THEN NS=3           <SPT>
2920 PRINT "{4SPACE,GREEN}X {CYAN}==> {GREEN}AUSG
   ABEMENUE{6SPACE}- {CYAN}==> {GREEN}SEITE ";V
   S;"{6SPACE}+ {CYAN}==> {GREEN}SEITE ";NS;            <8CT>
2930 DO                                                 <9D7>
2940 GET KEY A$: A=ASC(A$)                              <K3S>
2950 LOOP UNTIL (A=88 OR A=43 OR A=45)                  <QRD>
2960 IF A=88 THEN EXIT                                  <925>
2970 IF A=43 THEN W=W+1: ELSE W=W-1                     <T50>
2980 W=W-INT(W/3)*3                                     <OG6>
2990 IF W=0 THEN W=3                                    <GKI>
3000 LOOP                                               <049>
3010 RETURN                                             <O67>
3020 REM **** MONATSUEBERSICHT ****                     <1K6>
3030 DO                                                 <VFV>
3040 K=(MO-1)*5: IF D=1 THEN BEGIN                      <JTA>
3050 PRINT "{CLR,CYAN}" SPC(10);: PRINT USING "##
   #######";M$(MO);: PRINT " {PURPLE}";J$: CLOS
   E 1: CLOSE 6: OPEN 1,4,7: OPEN 6,4,6: PRINT#
   6,CHR$(24): CMD 1                                    <IV0>
3060 FOR I=1 TO 79: PRINT "-";: NEXT : PRINT : PR
   INT : PRINT                                          <0HC>
3070 BEND                                               <LUU>
3080 PRINT "{CLR,CYAN}" SPC(10);: PRINT USING "##
   #######";M$(MO);: PRINT " {PURPLE}";J$               <N5T>
3090 PRINT : PRINT SPC(10)"{GREEN}TAG{CYAN}" SPC(
   13);: FOR I=1 TO 5: L=K+I: IF S(L,1)<>0 THEN
    PRINT USING "##.";S(L,1);: ELSE PRINT "{3SP
   ACE}";                                               <0TO>
3100 PRINT SPC(7);: NEXT I                              <UMO>
3110 PRINT : PRINT : PRINT SPC(10)"{GREEN}KM-STAN
   D{CYAN}" SPC(5);: FOR I=1 TO 5: L=K+I: IF S(
   L,2)<>0 THEN PRINT USING "#######";S(L,2);:
   ELSE PRINT "{7SPACE}";                               <DDN>
3120 PRINT SPC(3);: NEXT I                              <S6G>
3130 PRINT : PRINT : PRINT SPC(10)"{GREEN}GEF.KM{
   CYAN}" SPC(7);: FOR I=1 TO 5: L=K+I: IF S(L,
   5)<>0 THEN PRINT USING "#######";S(L,5);: EL
   SE PRINT "{7SPACE}";                                 <IG1>
3140 PRINT SPC(3);: NEXT I                              <Q68>
3150 PRINT : PRINT : PRINT SPC(10)"{GREEN}GET.LIT
   ER{CYAN}" SPC(5);: FOR I=1 TO 5: L=K+I: IF S
   (L,3)<>0 THEN PRINT USING "###.##";S(L,3);:
   ELSE PRINT "{6SPACE}";                               <JTQ>
3160 PRINT SPC(4);: NEXT I                              <OQ0>
3170 PRINT : PRINT : PRINT SPC(10)"{GREEN}KOSTEN
   (DM){CYAN}" SPC(3);: FOR I=1 TO 5: L=K+I: IF
    S(L,4)<>0 THEN PRINT USING "###.##";S(L,4);
   : ELSE PRINT "{6SPACE}";                             <PRL>
3180 PRINT SPC(4);: NEXT I                              <MRO>
3190 PRINT : PRINT : PRINT SPC(10)"{GREEN}VERBR./
   100KM{CYAN}" SPC(2);: FOR I=1 TO 5: L=K+I: I
   F S(L,6)<>0 THEN PRINT USING "###.##";S(L,6)
   ;: ELSE PRINT "{6SPACE}";                            <RPO>
3200 PRINT SPC(4);: NEXT I                              <OQ0>
3210 PRINT : PRINT SPC(10)"{GREEN}IN LITER": PRIN
   T : PRINT SPC(10)"KOSTEN/LITER{CYAN}" SPC(2)
   ;: FOR I=1 TO 5: L=K+I: IF S(L,7)<>0 THEN PR
   INT USING "##.###";S(L,7);: ELSE PRINT "{6SP
   ACE}";                                               <8D1>
3220 PRINT SPC(4);: NEXT I                              <QQ8>
3230 PRINT : PRINT SPC(10)"{GREEN}IN DM"                <3FK>
3240 PRINT : PRINT : PRINT SPC(10)"{GREEN,CTRL+B}
   MONATSUEBERBLICK:"+CHR$(27)+"O": IF D=1 THEN
    PRINT : PRINT                                       <89A>
3250 PRINT : PRINT SPC(10)"{GREEN}GEF.KM{CYAN,4SP
   ACE}";: PRINT USING "#######";R(MO,1);: PRIN
   T "{GREEN,3SPACE}KOSTEN/LITER {CYAN}";: PRIN
   T USING "##.###";R(MO,3);                            <5E5>
3260 PRINT "{GREEN,4SPACE}VERBR./100KM {CYAN}";:
   PRINT USING "###.##";R(MO,2): IF D=1 THEN BE
   GIN                                                  <QVM>
3270 FOR I=1 TO 5: PRINT : NEXT : PRINT SPC(10)"N
   OTIZEN:": FOR I=1 TO 13: PRINT : NEXT : FOR
   I=1 TO 79: PRINT "-";: NEXT : PRINT : PRINT#
   1: PRINT#6,CHR$(36): CLOSE 1: CLOSE 6                <34N>
3280 BEND                                               <PRE>
3290 PRINT "{GREEN}";: FOR I=1 TO 78: PRINT "@";:
    NEXT : PRINT                                        <FOL>
3300 PRINT "{UP}"AU$;S2$;"{RED}@";: Z=2: PU=1: GO
   SUB 1290: IF F=2 THEN EXIT                           <D8K>
3310 IF B<13 AND B>0 THEN MO=B: ELSE MO=MO+1: IF
   MO=13 THEN MO=1                                      <SUB>
3320 LOOP                                               <G69>
3330 RETURN                                             <OE7>
```

Listing 1. (Schluß)

Komfortable Diskettenverwaltung

Bringen Sie Ordnung in Ihre Diskettensammlung mit Disk-Manager 128. Mit vielen Suchkriterien, einer sinnvollen Einteilung nach Programmarten und übersichtlicher Druckerausgabe wird die Archivierung Ihrer Programm-Disketten in Zukunft zu einem Kinderspiel.

Disk-Manager 128 arbeitet im 80-Zeichen-Modus des C128 und verarbeitet sowohl das 1571- als auch das 1541-Diskettenformat. Das Programm kann in vielen Variationen suchen, Listen drucken und Daten ändern. Es ist durch die komfortable Menüsteuerung leicht bedienbar. Das Programm besitzt eine universelle Druckeransteuerung und läuft mit praktisch jedem Drucker, es muß aber unter Umständen die Sekundäradresse in Zeile 150 (Listing 2) entsprechend angepaßt werden.



Bild 1. So sieht der Bildschirm aus, wenn Sie den Menüpunkt »DATEN ÄNDERN« aktivieren.

Bitte geben Sie Listing 1 (die Laderoutine) und Listing 2 (Disk-Manager 128) mit dem Checksummer 128 ein. Beachten Sie bitte dazu die Eingabehinweise auf Seite 158. Falls Sie auf das Titelbild verzichten wollen, können sie in Listing 1 die Zeilen 40 bis 270 weglassen. Interessierte finden außerdem eine genaue Beschreibung der einzelnen Routinen des Programms in Tabelle 1.

Bequeme Menüsteuerung

Nach dem Laden und Starten verlangt das Programm zunächst den Dateinamen der Archiv-Datei. Geben Sie entweder den Namen der Datei ein, die Sie laden wollen, oder den einer neu anzulegenden Datei. Legen Sie Ihre Datendiskette ein und drücken Sie <RETURN>. Wird die Datei auf der Diskette nicht gefunden, so bricht das Programm den Lesevorgang ab.

Wenn Sie eine neue Datei anlegen wollen, so beantworten Sie die Frage nach einer Neudatei mit <J> für »ja«. Sie befinden sich nun im Hauptmenü. Hier bieten sich die folgenden Wahlmöglichkeiten:

A – DATEN SUCHEN
B – DATEN EDITIEREN
C – DATEN DRUCKEN
D – DIRECTORY
E – DISKETTENKOMMANDO
F – PROGRAMM VERLASSEN

Zur Anwahl der einzelnen Menüpunkte drücken Sie bitte die entsprechende Buchstabentaste.

Viele Suchkriterien

Eine Stärke des Programms sind die umfangreichen Suchmöglichkeiten.

A – DATEN SUCHEN:
Nachdem Sie im Hauptmenü die Taste <A> gedrückt haben, befinden Sie sich im Untermenü »DATEN SUCHEN«. Hier haben Sie folgende Wahlmöglichkeiten:

A – PROGRAMMNAMEN
B – PROGRAMMLÄNGE
C – PROGRAMMART
D – PROGRAMMTYP
E – DISKETTEN
F – FREIE BLÖCKE
G – HAUPTMENUE

A – PROGRAMMNAMEN SUCHEN:
In diesem Programmteil können Sie nach beliebigen Programmnamen suchen. Sie haben dazu folgende Möglichkeiten: Entweder Sie geben den Namen vollständig – also mit allen Leerzeichen – ein, oder nur einen Teil des Namens und am Ende den üblichen Joker »*«. Der Joker bewirkt, daß alle Programme gesucht und angezeigt werden, die mit den vorangegangenen Zeichen beginnen. Ausgegeben werden Programm, Art des Programms (Spiel, Anwendung, etc.), Diskettenseite und Filetyp. Dies gilt für alle Punkte des »DATEN SUCHEN«-Menüs.
Beispiele:
»OSCAR« gibt nur das Programm »OSCAR« aus
»OS*« sucht alle Programme die mit »OS« beginnen
»*« gibt alle Programme aus
Nach Beendigung des Suchvorgangs gelangen Sie durch Drücken einer beliebigen Taste wieder in das Untermenü »DATEN SUCHEN«.

B – PROGRAMMLÄNGE SUCHEN:
Das Suchkriterium in diesem Unterprogramm ist die Programmlänge. Mit der ersten Eingabe gibt man die mittlere Suchlänge an, mit der zweiten die Abweichung nach unten und oben. Wird eine 0 als Suchlänge eingegeben, so erfolgt ein Rücksprung zum »DATEN SUCHEN«-Menü.
Beispiele:

Suchlänge	Abweichung	Wirkung
200	0	sucht alle Programme mit der Länge 200
200	10	sucht alle Programme mit einer Länge von 190 bis 210 Blöcken

C – PROGRAMMART SUCHEN:
Hier kann nach vorher definierten Programmarten (siehe dazu »PROGRAMMARTEN ÄNDERN«) gesucht werden.

Die gewünschte Programmart wird durch Eingabe der entsprechenden Nummer angewählt. Anschließend werden alle Programme dieser Art ausgegeben.

D - PROGRAMMTYP SUCHEN:
Funktioniert ähnlich wie »PROGRAMMART SUCHEN«, nur kann hier nach bestimmten Filetypen (PRG, SEQ, REL, etc.) gesucht werden.

E - DISKETTEN SUCHEN:
Hier werden alle Programme einer bestimmten Diskette ausgegeben. Geben Sie auf die Frage nach der Suchdiskette die gewünschte Diskettennummer an. Existiert diese Diskettennummer nicht, so verlangt das Programm eine Neueingabe. Ausgegeben werden alle Files beider Diskettenseiten und die jeweils freien Blöcke.

Sind mehr Einträge vorhanden als es die Fenstergröße zuläßt, so scrollen die Einträge nach oben. Durch Drücken einer beliebigen Taste kann das Scrollen angehalten und wieder fortgesetzt werden.

F - FREIE BLÖCKE SUCHEN:
Dieser Programmteil gibt die freien Blöcke der in der Datei gespeicherten Disketten aus. Dabei können Sie die Mindestanzahl der freien Blöcke, ab der eine Ausgabe erfolgen soll, angeben. Daraufhin werden alle Diskettenseiten die die freie Blockanzahl besitzen oder übersteigen, angezeigt. Sind mehr Disketten vorhanden als angezeigt werden können, so hält das Programm zunächst an und fährt nach einem Druck auf eine beliebige Taste wieder fort.

G - HAUPTMENUE:
Diese Funktion führt einen Rücksprung zum Hauptmenü aus.

B - DATEN EDITIEREN:
Auch der zweite Punkt des Hauptmenüs ist ausgesprochen komfortabel. Nachdem Sie im HAUPTMENUE die Taste <B> gedrückt haben befinden Sie sich im »DATEN EDITIEREN«-Menü. Hier gibt es folgende Wahlmöglichkeiten:

A - NACHLADE PRG MODUS AN/AUS
B - DATEN LESEN
C - DATEN ÄNDERN
D - DATEN EINFÜGEN
E - PROGRAMMARTEN ÄNDERN
F - PROGRAMMTYPEN ÄNDERN
G - DIRECTORY
H - DISKETTENKOMMANDO
I - HAUPTMENUE

A - NACHLADE PRG MODUS AN/AUS:
Nun zu einer besonderen Eigenschaft des Diskmanager 128; dem »NACHLADE PRG MODUS«, den Sie in diesem Menüpunkt aktivieren können.

Wenn dieser Modus eingeschaltet ist, werden Programmeinträge die im Directory zwischen zwei Trennmarkierungen (16 Minuszeichen) stehen beim Einlesen zusammengefaßt. Das heißt, die Blocks der einzelnen Files zwischen den beiden Markierungen werden addiert und unter dem Namen des ersten Files registriert.

Ist dieser Modus abgeschaltet, werden alle Einträge zwischen Trennmarkierungen als eigenständige Objekte betrachtet.

B - DATEN LESEN:
Hier können die Daten von zu verwaltenden Disketten gelesen werden. Geben Sie die Diskettennummer unter der die Diskette verwaltet werden soll, sowie die Diskettenseite ein und drücken Sie <RETURN>. Existieren schon Einträge mit dieser Nummer und Seite werden diese gelöscht und anschließend die neuen Daten gelesen.

Nach erfolgreichem Lesen gelangt man in das Untermenü »DATEN ÄNDERN« das zum Ändern der gelesenen Daten dient.

C - DATEN ÄNDERN (siehe Bild 1):
Nachdem Sie Nummer und Seite der zu ändernden Diskette eingegeben haben, können Sie - sofern nötig - die Anzahl der freien Blocks im Eintrag ändern.

Mit »A« gelangen Sie in einen Modus, in dem Sie nacheinander die verschiedenen Punkte wie Programmname, Länge und Art des Programms sowie Typ ändern können. Die einzelnen Änderungen müssen mit <RETURN> abgeschlossen werden.

Mit <SPACE> kommen Sie zum nächsten Eintrag, <L> löscht den gerade bearbeiteten.

Wichtiger Hinweis: Erstreckt sich ein Programm über mehrere Diskettenseiten, so ist es am besten, wenn Sie alle Teile unter dem gleichen Namen verwalten. Auf diese Weise werden diese zusammengefaßt und bei einem Suchvorgang als File ausgegeben, das sich über mehrere Diskettenseiten erstreckt.

Haben Sie jedoch verschiedene Versionen eines Programms, so geben Sie diesen verschiedene Namen, was schon durch das Anhängen eines »SPACE« geschieht.

D - DATEN EINFÜGEN
Mit diesem Unterprogramm ist es möglich, neue Einträge einzufügen. Es erscheint dabei als letzter Eintrag folgende Leermaske:

Name: NEUER EINTRAG Blöcke: 0 Art:KEIN EINTRAG Typ: ???

In der gleichen Weise wie bei »DATEN ÄNDERN« können Sie nun den Eintrag vornehmen.

Verwaltung der Files nach verschiedenen Progammarten

E - PROGRAMMARTEN ÄNDERN:
Hier können die vorgegebenen Programmarten verändert werden. Im oberen Teil des Bildschirms sehen Sie die aktuellen Programmarten. Wollen Sie eine davon ändern, so geben Sie die entsprechende Nummer ein. Nun können Sie einen neuen Text mit einer maximalen Länge von 16 Zeichen eingeben.

Wenn Sie für Programmart eine »0« eingeben, gelangen Sie wieder ins »DATEN EDITIEREN«-Menü, dies gilt auch für »PROGRAMMTYPEN ÄNDERN«.

F - PROGRAMMTYPEN ÄNDERN:
Dieses Unterprogramm unterscheidet sich nur dadurch von »PROGRAMMARTEN ÄNDERN«, daß hier die Programmtypen verändert werden können. Diese dürfen höchstens eine Länge von 3 Zeichen haben.

G - DIRECTORY:
Diese Funktion gibt das Directory der sich im Laufwerk befindlichen Diskette aus. Das Scrollen kann durch Drücken von <RUN/STOP> unterbrochen werden. Mit einer beliebigen Taste gelangen Sie wieder ins »DATEN EDITIEREN«-Menü

H - DISKETTENKOMMANDO:
Geben Sie das gewünschte Kommando direkt nach dem »<« ein. Zum Beispiel:
`<R:DATEI 2=DATEI 1`

I - HAUPTMENUE:
Mit dieser Funktion gelangt man ins Hauptmenü: Aber je nachdem, welche Funktionen des »DATEN EDITIEREN«-Menüs vorher angewählt wurden, wird vor dem Sprung zum Hauptmenü noch Verschiedenes ausgeführt:

Wenn Sie »PROGRAMMARTEN ÄNDERN« oder »PROGRAMMTYPEN ÄNDERN« angewählt hatten, werden vor

dem Sprung ins Hauptmenü noch die Daten gespeichert. Legen Sie dazu auf die Aufforderung die Datendiskette ein und drücken Sie <RETURN>. Bei der Speicherung werden die alten Daten überschrieben.

Hatten Sie »DATEN LESEN«, »DATEN ÄNDERN« oder »DATEN EINFÜGEN« angewählt, so wird ebenfalls gespeichert, aber vorher noch sortiert.

Bei den restlichen Punkten erfolgt ein direkter Sprung zum Hauptmenü.

Ausdruck übersichtlicher Listen

Nun zu einem sehr wichtigen Punkt: dem Ausdruck der Listen.

C – DATEN DRUCKEN:
Nachdem sie im Hauptmenü die Funktion »DATEN DRUCKEN« gewählt haben, fordert Se das Programm auf, das aktuelle Datum einzugeben. Anschließend können Sie mit den Tasten <A> bis <F> auswählen:
A - LISTE ALPHABETISCH
B - LISTE DISKETTEN
C - LISTE PROGRAMMART
D - LISTE PROGRAMMTYP
E - LISTE FREIE BLÖCKE
F - HAUPTMENUE

A - LISTE ALPHABETISCH:
Geben Sie bitte den ersten und den letzten zu druckenden Eintrag an. Weiters, ob sie Endlospapier <R> oder Einzelblätter <E> benutzen. Schalten Sie nun ihren Drucker ein und drücken Sie <RETURN>. Während des Druckvorganges werden auf dem Bildschirm die Programme, die ausgedruckt werden, angezeigt. Ist eine Seite vollgedruckt, wird je nachdem, ob sie <R> oder <E> gedrückt haben, ein Seitenvorschub gemacht oder ein neues Blatt verlangt. Sind alle Daten gedruckt, wird ein Form-Feed an den Drucker gesendet und Sie befinden sich wieder im »DATEN DRUCKEN«-Menü.

B - LISTE DISKETTEN:
Ist ähnlich dem Punkt »LISTE ALPHABETISCH«, nur daß Sie hier die Nummern der Disketten angeben müssen, zwischen welchen ausgedruckt werden soll.

C - LISTE PROGRAMMART:
Hier müssen Sie Beginn und Ende des Druckes durch die Nummern der jeweiligen Programmarten eingeben.

D - LISTE PROGRAMMTYP:
Geben Sie die entsprechenden Nummern der Filetypen ein, mit welchen die Liste beginnen, beziehungsweise enden soll.

E - LISTE FREIE BLÖCKE:
In diesem Programmteil müssen Sie die erste und letzte Nummer der Disketten angeben, von denen Sie die freien Blöcke ausgedruckt haben wollen.

F - HAUPTMENUE:
Diese Funktion führt einen Rücksprung zum Hauptmenü aus.

D – DIRECTORY:
Lädt das Directory.

E – DISKETTENKOMMANDO:
Dient zur Eingabe von Diskettenkommandos.

F – PROGRAMM VERLASSEN:
Diese Funktion führt einen Systemreset aus.

Sicherheitskopien der Dateien erstellen

In Listing 3 finden Sie noch ein Kopierprogramm mit dem Sie Sicherheitskopien der Systemdateien anlegen können. Bitte geben Sie das Listing mit dem Checksummer 128 ein (Eingabehinweise siehe Seite 159). Nach dem Laden und Starten geben Sie für »Quellname« den Namen ihrer Datei und für »Zielname« den Namen, unter dem diese neu abgespeichert werden soll, ein. Es ist auf jeden Fall empfehlenswert, sich eine Sicherheitskopie seiner Hauptdatei zu erstellen.

Anschließend nochmals der Hinweis: Alle, die genau wissen wollen, wie der Disk-Manager 128 arbeitet, finden in Tabelle 1 eine ausführliche Dokumentation.

(Michael Neumann/Martin Jobst)

Programmbeschreibung zu »Disk-Manager 128«

Zeile
130 – 140: Umschalten auf den 80-Zeichen-Bildschirm und den Fastmodus.
150: SA = Sekundäradresse des Druckers
153: »D« bestimmt die Maximalanzahl von Disketten (Vorder- und Rückseite). »P« ist die maximale Anzahl von Programmeinträgen. Ist ein Programm länger als eine Diskettenseite, so verwaltet Disk-Manager pro Seite und Programm einen Eintrag. Nach diesen beiden Variablen werden die Felder dimensioniert.
155: Fehlerbehebung ab Zeile 10500.
160 – 170: Dimensionierung der Felder. Das dreidimensionale Feld »DB« ist die Disketten-BAM. Der erste Parameter bestimmt die Diskettenseite (1=A;2=B). Der zweite Parameter bestimmt, ob die freien Blöcke oder die Belegung ausgegeben werden sollen (1=FREIE BLÖCKE, 2=BELEGUNG;) Wenn letzteres der Fall ist, bedeutet »DB=1«, daß die Seite belegt ist, »DB=0« das Gegenteil. Der dritte Parameter schließlich bestimmt die Diskettennummer. Beispiele:
»A=DB(1,1,10)«: In Variable »A« stehen die freien Blöcke der Diskette mit der Nummer 10, Seite A.
»A=DB(1,2,21)«: In Variable »A« steht, ob sich auf Diskette 21, Seite A Programme befinden (»A=1«: belegt) oder nicht (»A=0«: unbelegt).
Die Felder »NA$«, »BL«, »DI« ,»AR« und »TY« geben zum jeweiligen Eintrag den Namen, Blocklänge, Diskettenseite, Programmart und Programmtyp an. Aus »DI« läßt sich folgendermaßen die Diskette bzw. Seite bestimmen: Ist ihr Inhalt ungerade, so befindet sich dieses Programm auf einer A-Seite, ist er gerade, auf einer B-Seite. Die Nummer der Diskette läßt sich folgendermaßen berechnen: Diskette=»(DI(X)+1)/2«.
In »AL$« befinden sich die vorgegebenen Programmarten (bei Erstbenutzung). Die Felder »A« und »B« werden bei »DISKETTEN-DRUCKEN« erklärt.
180 – 260: Grundbelegung der Programmarten und Programmtypen.
270 – 330: Menümasken zum »DATEN EDITIEREN«-Menü.
340 – 710: Eingeben des Dateinamens und Laden der Datei. Die Eingabeprozedur (Zeile 390; Erklärung Zeile 10100) liefert einen String »E$« der Länge »L«, der aus ASCII-Zeichen besteht. Es werden nur Zeichen akzeptiert, die zwischen »A1« und »A2« liegen. In Zeile 440 wird ein Fehler bei der Eingabe abgefangen und entsprechend den Zeilen 490-500 verzweigt. Ab Zeile 520 wird die sequentielle Datei geladen.
720 – 930: Hauptmenü. Verzweigung gemäß Zeile 920. Zeile 910 führt einen Systemreset aus (SYS 65341).

Tabelle 1. Detailierte Erläuterungen zum Programm »Disk-Manager 128«

940 – 1140: »DATEN SUCHEN«- Menü. Die Variable »DM$« in Zeile 960 gehört zum Programm-Modul »Programmnamen Ausgeben«. Sie wird in Zeile 10240 erklärt.

1150 – 1260: Eingeben des Suchnamens mittels der Prozedur in Zeile 10100.

1270 – 1290: Variable »IC« gehört zur Prozedur in Zeile 10240. Variable »C« gibt an, ob am Schluß des Suchnamens ein »*« steht.

1300 – 1360: Mit Hilfe einer FOR-NEXT-Schleife werden die Namen von Anfang bis Ende ausgegeben, wobei »PZ« die Anzahl der Namen ist.

1370 – 1520: Nach der Eingabe werden Suchlänge und Abweichung in die Variablen »LA« und »DD« übergeben.

1530 – 1620: Damit auch Programme erfaßt werden können, die mehr als einen Eintrag besitzen, werden innerhalb der Schleife die Blöcke von gleichnamigen Einträgen zusammengezählt, verglichen und die Laufvariable »I« wieder zurückgesetzt.

1630 – 1880: Hier werden die möglichen Programmarten ausgegeben und die Suchart eingegeben. Danach werden mit Hilfe einer Schleife die Programmarten mit der Suchart verglichen und gegebenenfalls ausgegeben.

1890 – 2150: Dieses Modul ist analog aufgebaut zu Programmart suchen, nur wird hier nicht das Eingabemodul aufgerufen, sondern mit »CHR$(15)« ein blinkender Cursor erzeugt und mit »CHR$(143)« wieder ausgeschaltet.

2160 – 2240: Modul »DISKETTEN SUCHEN«. Eingabe der Suchdiskette.

2245: Bei einer Fehlermeldung erfolgt die Kontrolle, ob wenigstens eine Diskettenseite der Suchdiskette belegt ist.

2250 – 2470: Ausgabe der Einträge der A- und B-Seite sowie der Anzahl der freien Blöcke. Wenn eine Seite nicht belegt ist, dann werden automatisch 664 freie Blöcke angenommen.

2480 – 2620: Programmmodul »FREIE BLÖCKE SUCHEN«. Eingabe der Mindestblockzahl.

2630 – 2810: Ausgabe der ermittelten Daten auf drei Windows, die nacheinander angezeigt werden.

2820 – 2970: Datumseingabe. Das Datum steht am Ende der Variable »DA$«.

2980 – 3160: »DATEN DRUCKEN«-Menü.

3170 – 3330: Modul »LISTE ALPHABETISCH«. Eingabe der Druckgrenzen »G1$« und »G2$«, sowie des Druckmodus »DM$« für Einzelblätter oder Endlospapier.

3340 – 3370: Drucken des Seitenkopfes.

3380 – 3440: Drucken von Überschrift, Seitenzahl, Datum und Dateiname.

3460 – 3510: Prüfung, ob der Eintrag innerhalb der Grenzen liegt und dessen Ausgabe mit Hilfe der Zeile 10250.»DM$« wird außerdem benötigt, um eine Ausgabe der kompletten Druckzeile auf dem Bildschirm zu verhindern. Deshalb wird »DM$« in Zeile 960 gelöscht.

3520: Hier wird geprüft, ob schon mehr als 60 Zeilen auf einem Blatt gedruckt sind.

3540 – 3630: Je nachdem, ob der Inhalt von »DM$« »R« oder »E« ist, werden entweder 7 Line-Feeds oder ein Form-Feed gesendet.

3640 – 3880: »LISTE DISKETTEN« drucken. Eingabe der Grenzen »DL« und »DH«, sowie der Variablen »DM$«. Ausgabe des Seitenkopfes.

3900 – 3950: Durchsuchen des Feldes »DI« (der jeweiligen Diskette) nach den gewünschten Einträgen und Ablegen der gefundenen Stellen in den Feldern »A« und »B«.

3960 – 3970: Hier wird geprüft, ob alle Einträge der Diskette (beide Seiten) auf das eingespannte Blatt (63 Zeilen) passen.

3980 – 4390: Ausgeben des Disketteninhalts auf den Drucker und anschließender Seitenvorschub (siehe Zeilen 3540 – 3630).

4400 – 5110: »LISTE PROGRAMMART«. Im wesentlichen identisch zu »LISTE ALPHABETISCH«, nur daß bei Beginn der nächsten Programmart zunächst deren Name zusammen mit einem neuen Kopf ausgedruckt wird.

5120 – 5720: »LISTE PROGRAMMTYP«. Unterscheidet sich nur durch das Suchobjekt von »LISTE PROGRAMMART«.

5730 – 6150: »LISTE FREIE BLÖCKE« drucken. Eingabe der Grenzen »G1« und »G2« sowie der Papierart »DM$«. Drucken der Daten aus dem Feld »DB«, geordnet nach Diskettennummer.

6160 – 6390: »DATEN EDITIEREN«-Menü.

6400 – 6420: »NACHLADE PRG MODUS«. Je nachdem, ob dieser aktiviert ist oder nicht, ist der Inhalt von»TR« gleich »1« oder »0«.

6430 – 6570: Zunächst wird eine Unterroutine zur Eingabe von Diskette und Seite aufgerufen (ab Zeile 9930). Dieses liefert die Seite »S«, die Diskettennummer »DN«, sowie »DK«. »DK« belegt das Feld »DI«, um auf Diskettennummer und Seite des jeweiligen Eintrages rückschließen zu können (Berechnung siehe Zeile 10080).
Daraufhin werden alle auf der Diskette bereits existierenden Neueinträge, gelöscht. Das heißt, sie werden mit dem Namen »ZZZZ« versehen und später aussortiert. Wenn die neue Diskettennummer »DN« größer als »DZ« ist, wird »DN« übernommen und diese Diskettenseite als belegt in der Diskettenbam aufgenommen.

6580 – 7020: Einlesen der Daten aus dem Directory. Diese Routine liest die Daten aus dem Directory der Diskettenlaufwerk 1541 oder 1571 ein. Die so ermittelten Daten werden in die Felder »NA$«, »BL«, »DI«, »AR« und »TY« übernommen. Bei »TR=0« werden die Blöcke solange zusammengezählt, bis eine gestrichelte Linie (aus Minuszeichen) der nächste zu lesende Eintrag ist.

7030 – 7040: Übernehmen der freien Blöcke in die Bam und direkter Sprung zum »DATEN ÄNDERN«-Menü.

7050 – 7130: »DATEN ÄNDERN«-Modul. Eingabe der Diskettennummer und der Seite.

7140: Ansprungadresse für »DATEN LESEN« und »DATEN EINFÜGEN«.

7150 – 7290: Aufbau des »DATEN ÄNDERN«-Menüs.

7300 – 8030: FOR-NEXT-Schleife mit Laufvariable »IC« zum Durchlaufen aller Einträge.
Zunächst werden im Menüfenster alle aktuellen Einträgen ausgegeben (Zeilen 7320 – 7360). Anschließend können die freien Blöcke geändert werden. In der rechten oberen Ecke wird nun mit Hilfe von »CHR$(15)« ein blinkendes »FREIE BLÖCKE:« erzeugt, und ein neues Eingabemodul (ab Zeile 9520) verlangt die Eingabe.
Die Bedeutungen der Parameter sind folgende: : »A« ist die linke, »B« die rechte X-Koordinate. »G« gibt die Zeile, in der das einzeilige Window erscheint, an. »C« ist die maximale Länge des Eingabestrings und »D« beziehungsweise »E« sind die ASCII-Grenzen der Eingabetasten (weil manchmal nur Zahlen eingegeben werden dürfen). »D$« ist der alte String, der wieder übernommen wird, wenn nur <RETURN> gedrückt wird.
In den Zeilen 7570 – 8010 werden alle Daten editierbar gemacht, indem ein roter Rahmen (Zeilen 270 – 330) über den schwarzen Rahmen gelagert und die Eingaberoutine aufgerufen wird.
Nur beim Ändern von Programmarten und Programmtypen wird ein eigenes Eingabemodul benutzt.

8040 – 8070: Vor der FOR-NEXT-Schleife wurde die Variable »LI=0« gesetzt. Drücken sie nun bei allen Einträgen der Seite <L>, so werden nicht nur die Namen mit »ZZZZ« belegt, sondern auch die Seite aus der Bam gelöscht (»DB(2,S,DN)=0«), indem »LI=0« bleibt. Drücken sie dagegen nur einmal <SPACE> oder <A> so wird »LI=1« gesetzt und die Seite wird nicht gelöscht.

8080 – 8160: »DATEN EINFÜGEN«. Nach Eingabe der Diskettennummer wird ein neuer Eintrag erschaffen, indem »PZ« um 1 erhöht, und die Parameter belegt werden. Danach befinden sie sich wieder in der »DATEN ÄNDERN«-Maske.

8170 – 8350: »PROGRAMMARTEN ÄNDERN«. Zuerst werden die aktuellen Programmarten gezeigt. Dann wird mit dem Modul ab Zeile 10010 die zu ändernde Art eingegeben. Wird »0« eingegeben, erfolgt ein Rücksprung zum »DATEN EDITIEREN«-Menü. Anschließend Eingabe der neuen Programmart.

8360 – 8550: »PROGRAMMTYPEN ÄNDERN«.Analog zu »PROGRAMMARTEN ÄNDERN«.

8560 – 8580: Je nachdem welche(r) Punkt(e) vorher gewählt wurde(n), wird vor dem Sprung zum Hauptmenü noch eine andere Funktion ausgeführt.
Diese wird in der Variable »SI« festgelegt: Ist »SI« gleich 0 (bei »DIRECTORY«, »DISKETTENKOMMANDO«) wird ein direkter Sprung zum Hauptmenü ausgeführt. Bei SI=1 (bei »PROGRAMMARTEN ÄNDERN«, »PROGRAMMTYPEN ÄNDERN) wird vorher gespeichert. Ist »SI« gleich 2 (bei den restlichen Punkten) wird zusätzlich noch sortiert.

8590 – 8700: Berechnung der notwendigen Rechenschritte »SZ« sowie des Bildschirmaufbaus.
8710 – 9200: Sortierverfahren Heap-Sort (Baumstrukturiertes sortieren) Dieses Sortierverfahren ist besonders schnell, denn im Gegensatz zu anderen Verfahren die »N*N« Zeiteinheiten benötigen, kommt dieses Verfahren mit »N*LOG(N)« Zeiteinheiten aus. (N=Anzahl der zu sortierenden Elemente)
9210 – 9230: Aussortieren der gelöschten Einträge (Name: ZZZZ).
9240 – 9510: Löschen der alten Datei und Abspeichern der neuen mit eventuellem Abbruch bei einem Diskettenfehler.
9520 – 9660: Eingabemodul I. Die »DO-LOOP-Schleife« führt je nach gedrückter Tastenkombination verschiedene Funktionen aus. »I1« gibt an, wieviele richtige Zeichen eingegeben wurden. Die »IF-Zeilen« dienen zum Ausschließen von Falscheingaben. Bei Drücken von <DEL> wird der String um 1 vermindert. Wenn nur <RETURN> eingegeben wird, erfolgt die Ausgabe von D$.
9730 – 9830: »DISKETTENKOMMANDO«. Eingabe eines Diskettenkommandos durch die Routine ab Zeile 10010 und Senden an die Floppy.
9840 – 9920: »DIRECTORY«. Die Directory kann durch Drücken von <RUN/STOP> bei »D=1« angehalten werden (Zeile 10520) und setzt danach wieder fort.
9930 – 10090: Angabe von Diskettennummer und Seite.
10100 – 10230: Eingabemodul II. Funktioniert ähnlich wie Eingabe I, jedoch wird der String »E$« direkt an der Stelle ausgegeben, an der der Cursor steht.
10240 – 10390: Berechnung von Blocklänge, Seite und Nummer jener Programme, die über eine Seite hinausgehen).
10390 – 10440: Berechnung von Seite und Nummer bei allen anderen Programmen.
10450 – 10490: Wenn »DM$ < > " "« (bei der Druckerausgabe), steht die Druckzeile in »D$«, ansonsten wird die Zeile in einem Window ausgegeben.
10500 – 10540: Fehlerbeseitigung bei Directory- und Druckerausgabe.

Tabelle 1. Detailierte Erläuterungen zum Programm »Disk-Manager 128« (Schluß)

```
10 REM DISK-MANAGER'LOADER'                           <A99>
20 FAST : GRAPHIC 5                                   <UNT>
30 COLOR 6,16: COLOR 5,1                              <JUO>
40 PRINT "{CLR,7DOWN}"                                <444>
50 PRINT "{3SPACE,RVSON,5SPACE}¥{RVOFF} {RVSON,2S
   PACE,RVOFF} {RVSON}£{4SPACE}¥{RVOFF} {RVSON,2S
   PACE,RVOFF} {RVSON}£ {RVOFF}£";                    <D5N>
60 PRINT "{4SPACE,RVSON,2SPACE}¥£{2SPACE,RVOFF} {
   RVSON}£{4SPACE}¥{RVOFF} {RVSON,2SPACE,RVOFF,2S
   PACE,RVSON,2SPACE,RVOFF} {RVSON}£{4SPACE}¥{RVO
   FF}";                                              <11B>
70 PRINT " {RVSON}£{4SPACE}¥{RVOFF} {RVSON,6SPACE
   ,RVOFF} {RVSON,5SPACE}¥{RVOFF}"               "    <BL7>
80 PRINT "{3SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVSON
   ,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF} {RVSON,2SP
   ACE,RVOFF,5SPACE,RVSON,2SPACE,RVOFF,RVSON}£ {R
   VOFF}£ ";                                          <P11>
90 PRINT "{4SPACE,RVSON,2SPACE,RVOFF}¥£{RVSON,2SP
   ACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,RVSON,2S
   PACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,RVSON,2
   SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,RVSON,
   2SPACE,RVOFF}";                                    <T09>
100 PRINT " {RVSON,2SPACE,RVOFF,5SPACE,RVSON,2SPA
    CE,RVOFF,5SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVS
    ON,2SPACE,RVOFF}"                                 <AG4>
110 PRINT "{3SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVSO
    N,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF} {RVSON,2
    SPACE,RVOFF,5SPACE,RVSON,3SPACE,RVOFF}£{2SPAC
    E}";                                              <VA3>
120 PRINT "{4SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVSO
    N,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,RV
    SON,2SPACE,RVOFF} {RVSON,2SPACE}¥{RVOFF} {RVS
    ON,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,R
    VSON,2SPACE,RVOFF}";                              <3GP>
130 PRINT " {RVSON,2SPACE,RVOFF,5SPACE,RVSON,2SPA
    CE,RVOFF,5SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVS
    ON,2SPACE,RVOFF}"                                 <6HK>
140 PRINT "{3SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVSO
    N,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF} ¥{RVSON,
    4SPACE}¥{RVOFF} {RVSON,3SPACE,RVSON}¥{RVOFF,2
    SPACE}";                                          <M3K>
150 PRINT " {RVSON,2SPACE,RVOFF} {RVSON,2SPACE,RV
    OFF,2SPACE,RVSON,2SPACE,RVOFF} {RVSON,6SPACE,
    RVOFF} {RVSON,2SPACE,RVOFF}¥{RVSON}¥{2SPACE,R
    VOFF} {RVSON,6SPACE,RVOFF}";                      <81F>
160 PRINT " {RVSON,2SPACE,RVOFF} {RVSON,3SPACE,RV
    OFF} {RVSON,6SPACE,RVOFF} {RVSON,5SPACE,RVOFF
    }£"                                               <L61>
170 PRINT "{3SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVSO
    N,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,5SPACE,RV
    SON,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF}¥{RVSON
    } ¥{RVOFF} ";                                     <B5E>
180 PRINT "{4SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVSO
    N,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,RV
    SON,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF} ¥{RVSO
    N,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,RV
    SON,2SPACE,RVOFF}";                               <95M>
190 PRINT " {RVSON,2SPACE,RVOFF,2SPACE,RVSON,2SPA
    CE,RVOFF} {RVSON,2SPACE,RVOFF,5SPACE,RVSON,3S
    PACE}¥"                                           <TLH>
200 PRINT "{3SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVSO
    N,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,5SPACE,RV
    SON,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF} ¥{RVSO
    N} ¥{RVOFF}";                                     <9I6>
210 PRINT "{4SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVSO
    N,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,RV
    SON,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,
    RVSON,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPAC
    E,RVSON,2SPACE,RVOFF}";                           <N69>
220 PRINT " {RVSON,2SPACE,RVOFF,2SPACE,RVSON,2SPA
    CE,RVOFF} {RVSON,2SPACE,RVOFF,5SPACE,RVSON,2S
    PACE,RVOFF}¥{RVSON} ¥{RVOFF}"                     <9E9>
230 PRINT "{3SPACE,RVSON,5SPACE,RVOFF}£ {RVSON,2S
    PACE,RVOFF} ¥{RVSON,4SPACE,RVOFF}£{RVOFF} {RV
    SON,2SPACE,RVOFF,2SPACE,RVSON,2SPACE,RVOFF}";     <J10>
240 PRINT "{4SPACE,RVSON,2SPACE,RVOFF,2SPACE,RVSO
    N,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,RV
    SON,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPACE,
    RVSON,2SPACE,RVOFF} {RVSON,2SPACE,RVOFF,2SPAC
    E,RVSON,2SPACE,RVOFF}";                           <NI9>
250 PRINT " ¥{RVSON,4SPACE,RVOFF}£ {RVSON,6SPACE,
    RVOFF} {RVSON,2SPACE,RVOFF} ¥{RVSON} ¥{RVOFF}
    "                                                 <31G>
260 PRINT TAB(35)"{2DOWN}VERSION V1.0"                <3F3>
270 PRINT TAB(23)"{DOWN}GESCHRIEBEN VON MICHAEL N
    EUMANN IN 1987"                                   <VQJ>
300 DLOAD "DISK.MAIN"                                 <O1G>
```

Listing 1. Die Laderoutine für »Disk-Manager 128«. Bitte mit dem Checksummer 128 (Seite 158) eingeben.

```
10 REM DISKMANAGER.MAIN                               <A99>
20 REM ***************************************        <AHA>
30 REM *                                     *        <APB>
40 REM *         DISK - MANAGER V1.0         *        <B1C>
50 REM *                                     *        <B9D>
60 REM *          GESCHRIEBEN VON :          *        <5SN>
70 REM *                                     *        <BPF>
80 REM *           MICHAEL NEUMANN           *        <810>
90 REM *          OSTBURGER WEG 79D          *        <891>
100 REM *            1000 BERLIN 47            *      <IIA>
110 REM *                                      *      <IQB>
120 REM ***************************************       <I28>
130 GRAPHIC 5,1                                       <32R>
140 FAST                                              <1G3>
150 SA=0 : REM HIER SETZEN SIE DIE SEKUNDAERADRES
    SE IHRES DRUCKERS EIN                             <89B>
153 D=100: P=300                                      <IKO>
155 TRAP 10500                                        <0PA>
160 DIM DB(2,2,D): DIM NA$(P): DIM BL(P): DIM DI(
    P): DIM AR(P): DIM TY(P)                          <AJO>
170 DIM AL$(19): DIM A(144): DIM B(144)               <8JO>
180 AL$(1)="ACTION-SPIEL": AL$(2)="STRATEGIE-SPIE
    L": AL$(3)="GESCHICKL.-SPIEL"                     <D2L>
190 AL$(4)="ADVENTURE-SPIEL": AL$(5)="SPORT-SPIEL
    ": AL$(6)="SIMULATOR-SPIEL"                       <E4I>
200 AL$(7)="SPIEL": AL$(8)="TEXTVERARBEITUNG": AL
    $(9)="DATEIVERWALTUNG"                            <KGT>
210 AL$(10)="KOPIERPROGRAMM": AL$(11)="DISKPROGRA
    MM": AL$(12)="MATHEMATIK PRG"                     <FUH>
220 AL$(13)="MUSIK PRG": AL$(14)="MUSIK DEMO": AL
    $(15)="GRAFIK PRG"                                <ITT>
230 AL$(16)="GRAFIK DEMO": AL$(17)="SPRACHE": AL$
    (18)="ANWENDUNG"                                  <FD5>
240 AL$(19)="KEIN EINTRAG"                            <PLD>
250 TL$(1)="PRG": TL$(2)="SEQ": TL$(3)="REL"          <K7B>
260 TL$(4)="USR": TL$(5)="DEL": TL$(6)="???": TL$
    (7)="???"                                         <OFH>
270 MR$(1)="{RED}UCCCCCCCCCCCCCCCCCCIB{18SPACE}BJ
```

Listing 2. Das komfortable Diskettenverwaltungsprogramm »Disk-Manager 128«. Bitte mit dem Checksummer (Seite 128) eingeben.

```
     CCCCCCCCCCCCCCCCCCCK"                              <4DK>
280 MS$(1)="{BLACK}UCCCCCCCCCCCCCCCCCCCB{18SPACE}
    BJCCCCCCCCCCCCCCCCCCCF"                             <4SE>
290 MR$(2)="{RED}UCCCCCCIB{6SPACE}BJCCCCCCK"            <IO3>
300 MS$(2)="{BLACK}RCCCCCCRB{6SPACE}BFCCCCCCF"          <P1E>
310 MS$(3)="{BLACK}RCCCCCCCCCCCCCCCCCCCRB{18SPACE}
    BFCCCCCCCCCCCCCCCCCCCF"                             <CSN>
320 MR$(4)="{RED}UCCCCCIB{5SPACE}BJCCCCCK"              <13Q>
330 MS$(4)="{BLACK}RCCCCCIB{5SPACE}BFCCCCCK"            <BEA>
340 COLOR 6,16: COLOR 5,1                               <IIS>
350 DO                                                  <JMF>
360 PRINT "{CLR,DOWN}BITTE GEBEN SIE IHREN DATEIN
    AMEN AN !"                                          <IRS>
370 FOR I=1 TO 80: PRINT "£";: NEXT I                   <7JH>
380 PRINT "{DOWN}DATEI - NAME : ";                      <UHS>
390 L=16: A1=32: A2=95: GOSUB 10100                     <U5H>
400 LOOP WHILE RIGHT$(E$,1)="*"                         <EOD>
410 DN$=E$                                              <MUD>
420 PRINT : PRINT "{DOWN}SUCHEND"                       <S9H>
430 DOPEN #1,(DN$)                                      <PEN>
440 IF DS<>0 THEN BEGIN                                 <BRS>
450 DCLOSE #1                                           <VL5>
460 PRINT "LESEFEHLER !!!"                              <ASE>
470 PRINT "{DOWN}NEUE DATEI (J/N) ?"                    <33U>
480 DO : GET KEY A$: LOOP UNTIL A$="J" OR A$="N"        <II2>
490 IF A$="N" THEN GOTO 350                             <KBL>
500 IF A$="J" THEN GOTO 720                             <9H2>
510 BEND                                                <MUU>
520 PRINT "LESEND"                                      <9GU>
530 FOR A=1 TO 6                                        <VE0>
540 INPUT#1,TL$(A)                                      <8N3>
550 NEXT A                                              <IU0>
560 FOR A=1 TO 18                                       <T4Q>
570 INPUT#1,AL$(A)                                      <CPJ>
580 NEXT A                                              <QJ0>
590 INPUT#1,DZ                                          <L0C>
600 FOR A=1 TO DZ                                       <C6U>
610 INPUT#1,DB(1,1,A)                                   <3G9>
611 INPUT#1,DB(1,2,A)                                   <0O4>
612 INPUT#1,DB(2,1,A)                                   <F1R>
613 INPUT#1,DB(2,2,A)                                   <C9M>
620 NEXT A                                              <QV0>
630 INPUT#1,PZ                                          <S0O>
640 FOR A=1 TO PZ                                       <D6A>
650 INPUT#1,NA$(A)                                      <ALA>
660 INPUT#1,TY(A)                                       <2NA>
670 INPUT#1,BL(A)                                       <JGE>
680 INPUT#1,DI(A)                                       <2TB>
690 INPUT#1,AR(A)                                       <RVF>
700 NEXT A                                              <QV0>
710 DCLOSE #1                                           <V55>
720 WINDOW 0,0,79,24                                    <323>
725 TRAP 10500                                          <59A>
730 COLOR 6,16: COLOR 5,1                               <TJ0>
740 PRINT "{CLR}"                                       <R9M>
750 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCI"           <JVD>
760 PRINT TAB(25)"B{3SPACE}DISK-MANAGER V1.0{4SPA
    CE}B"                                               <8DB>
770 PRINT TAB(25)"B{24SPACE}B"                          <5QR>
780 PRINT TAB(25)"B{7SPACE}HAUPTMENUE{7SPACE}B"         <II6>
790 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCK"           <VVF>
800 PRINT TAB(27)"{DOWN}A : DATEN SUCHEN"               <2PQ>
810 PRINT TAB(27)"{DOWN}B : DATEN EDITIEREN"            <GQG>
820 PRINT TAB(27)"{DOWN}C : LISTEN DRUCKEN"             <4JE>
830 PRINT TAB(27)"{DOWN}D : DIRECTORY"                  <70C>
840 PRINT TAB(27)"{DOWN}E : DISKETTENKOMMANDO"          <DP7>
850 PRINT TAB(27)"{DOWN}F : PROGRAMM VERLASSEN"         <DCE>
860 PRINT TAB(29)"{2DOWN}TASTE DRUECKEN"                <GC5>
870 DO                                                  <7KV>
880 GET KEY A$                                          <GKD>
890 IF ASC(A$)<71 AND ASC(A$)>64 THEN EXIT              <GQF>
900 LOOP                                                <RB9>
910 IF A$="F" THEN SYS 65341                            <TTQ>
920 ON  ASC(A$)- 64 GOSUB 940,6160,2820,9840,9730       <BIN>
930 GOTO 720                                            <QQ8>
940 REM DATEN SUCHEN                                    <NIU>
950 WINDOW 0,0,79,24                                    <0EE>
960 PRINT "{CLR,DOWN}": DM$=""                          <KNK>
970 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCI"           <KFD>
980 PRINT TAB(25)"B{6SPACE}DATEN SUCHEN{6SPACE}B"       <6EH>
990 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCK"           <OVF>
1000 PRINT TAB(27)"{DOWN}A : PROGRAMM-NAMEN"            <9MS>
1010 PRINT TAB(27)"{DOWN}B : PROGRAMM-LAENGEN"          <POE>
1020 PRINT TAB(27)"{DOWN}C : PROGRAMM-ART"              <JLG>
1030 PRINT TAB(27)"{DOWN}D : PROGRAMMTYP"               <0LS>
1040 PRINT TAB(27)"{DOWN}E : DISKETTEN"                 <1DK>
1050 PRINT TAB(27)"{DOWN}F : FREIEN BLOECKEN"           <BTN>
1060 PRINT TAB(27)"{DOWN}G : HAUPTMENUE"                <2FT>
1070 PRINT TAB(29)"{2DOWN}TASTE DRUECKEN"               <D5G>
1080 DO                                                 <SFJ>
1090 GET KEY A$                                         <5OD>
1100 IF ASC(A$)<72 AND ASC(A$)>64 THEN EXIT             <J61>
1110 LOOP                                               <C1P>
1120 IF A$="G" THEN RETURN                              <VIA>
1130 ON  ASC(A$)- 64 GOSUB 1150,1370,1630,1890,21
     60,2480                                            <4GH>
1140 GOTO 940                                           <66I>
1150 REM PROGRAMM-NAMEN SUCHEN                          <2SB>
1160 PRINT "{CLR,DOWN}"                                 <RAF>
1170 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCCI"         <1BD>
1180 PRINT TAB(25)"B{2SPACE}PROGRAMMNAMEN SUCHEN{
     2SPACE}B"                                          <LBH>
1190 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCCK"         <DRF>
1200 PRINT "{DOWN,2SPACE}NAME{13SPACE}BLOECKE{6SP
     ACE}PRORAMMART{9SPACE}DISKETTENSEITE{5SPACE}
     FILETYP"                                           <USR>
1210 FOR I=1 TO 80: PRINT "D";: NEXT I                  <7NI>
1220 PRINT "{13DOWN}"                                   <3D4>
1230 FOR I=1 TO 80: PRINT "D";: NEXT I                  <5NI>
1240 WINDOW 0,23,79,24                                  <1GR>
1250 PRINT "{CLR} SUCH - NAME : ";                      <JC8>
1260 L=16: A1=32: A2=95: GOSUB 10100                    <DLN>
1270 PRINT "{CLR}": IC=0: C=0                           <VON>
1280 T=LEN(E$)                                          <KBR>
1290 IF RIGHT$(E$,1)="*" THEN C=1                       <ANK>
1300 FOR I=1 TO PZ                                      <7M9>
1310 IF (LEFT$(NA$(I),T-1)+"*"=E$ AND C=1 AND NA$
     (I)<>"") OR NA$(I)=E$ THEN GOSUB 10250             <38D>
1320 NEXT I                                             <DH4>
1330 WINDOW 0,23,79,24                                  <0OU>
1340 PRINT "{CLR}";TAB(23)"KEINE WEITEREN EINTRAE
     GE VORHANDEN";                                     <SBP>
1350 GET KEY A$                                         <78D>
1360 RETURN                                             <BK7>
1370 REM PROGRAMM-LAENGEN SUCHEN                        <3CD>
1380 PRINT "{CLR,DOWN}"                                 <UAF>
1390 PRINT TAB(24)"UCCCCCCCCCCCCCCCCCCCCCCCCCCI"        <JO7>
1400 PRINT TAB(24)"B{2SPACE}PROGRAMMLAENGEN SUCHE
     N{2SPACE}B"                                        <VVP>
1410 PRINT TAB(24)"JCCCCCCCCCCCCCCCCCCCCCCCCCCK"        <BC5>
1420 PRINT "{DOWN,2SPACE}NAME{13SPACE}BLOECKE{6SP
     ACE}PRORAMMART{9SPACE}DISKETTENSEITE{5SPACE}
     FILETYP"                                           <EIR>
1430 FOR I=1 TO 80: PRINT "D";: NEXT I                  <9NI>
1440 PRINT "{13DOWN}"                                   <B84>
1450 FOR I=1 TO 80: PRINT "D";: NEXT I                  <FNI>
1460 WINDOW 0,23,79,24                                  <20L>
1470 PRINT "{CLR} SUCH - LAENGE (BLOECKEN) : ";         <JPD>
1480 L=4: A1=48: A2=57: GOSUB 10100: LA=VAL(E$)         <RIR>
1485 IF LA=0 THEN RETURN                                <9SL>
1490 PRINT "{CLR} +/- LAENGE (BLOECKEN) : ";            <MDO>
1500 GOSUB 10100: IF VAL(E$)>LA THEN GOTO 1470          <VC9>
1510 DD=VAL(E$)                                         <4B8>
1520 PRINT "{CLR}": IC=0                                <8G4>
1530 FOR I=1 TO PZ                                      <8M9>
1540 BL=0: CI=0                                         <U6A>
1550 DO : BL=BL+BL(I): I=I+1: CI=CI+1: LOOP WHILE
      NA$(I-1)=NA$(I)                                   <8NB>
1560 I=I-CI: IF CI>1 AND ((LA+DD)<BL OR (LA-DD)>B
     L) THEN I=I+CI-1                                   <TO7>
1570 IF (LA+DD)>=BL AND (LA-DD)<=BL THEN GOSUB 10
     250                                                <AES>
1580 NEXT I                                             <TN4>
1590 WINDOW 0,24,79,24                                  <PFO>
1600 PRINT "{CLR}";TAB(23)"KEINE WEITEREN EINTRAE
     GE VORHANDEN";                                     <C5P>
1610 GET KEY A$                                         <4OD>
1620 RETURN                                             <887>
1630 REM PROGRAMM-ART SUCHEN                            <0S3>
1640 PRINT "{CLR,DOWN}"                                 <TAF>
1650 PRINT TAB(26)"UCCCCCCCCCCCCCCCCCCCCCCCI"           <0MN>
1660 PRINT TAB(26)"B{2SPACE}PROGRAMMART SUCHEN{2S
     PACE}B"                                            <4FJ>
1670 PRINT TAB(26)"JCCCCCCCCCCCCCCCCCCCCCCCK"           <3UL>
1680 PRINT "{DOWN,2SPACE}NAME{13SPACE}BLOECKE{6SP
     ACE}PRORAMMART{9SPACE}DISKETTENSEITE{5SPACE}
     FILETYP"                                           <USR>
1690 FOR I=1 TO 80: PRINT "D";: NEXT I                  <7NI>
1700 PRINT "{DOWN,22SPACE}MOEGLICHE PROGRAMMARTEN
     "                                                  <NRJ>
1710 PRINT "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"           <V6A>
1720 FOR I=1 TO 6                                       <VM1>
1730 PRINT " (";I;") ";TAB(8)AL$(I);TAB(24)" (";I
     +6;") ";TAB(32)AL$(I+6);TAB(48)" (";I+12;")
     ";TAB(56)AL$(I+12)                                 <0HV>
1735 NEXT I                                             <HQK>
1740 PRINT TAB(48)" ( 19 ) KEIN EINTRAG"                <PK1>
1750 PRINT "{3DOWN}"                                    <JLK>
1760 FOR I=1 TO 80: PRINT "D";: NEXT I                  <ANI>
1770 WINDOW 0,23,79,24                                  <3OI>
1780 PRINT "{CLR} SUCH - ART : ";                       <OI0>
1790 L=2: A1=48: A2=57: GOSUB 10100                     <RHL>
1800 IF VAL(E$)>19 OR VAL(E$)=0 THEN RETURN             <F6C>
1810 PRINT "{CLR}": IC=0                                <0B7>
1820 FOR I=1 TO PZ                                      <JM9>
1830 IF VAL(E$)=AR(I) THEN GOSUB 10250                  <CEO>
1840 NEXT I                                             <D14>
1850 WINDOW 0,23,79,24                                  <40E>
1860 PRINT "{CLR}";TAB(23)"KEINE WEITEREN EINTRAE
     GE VORHANDEN";                                     <SVP>
1870 GET KEY A$                                         <28D>
1880 RETURN                                             <947>
1890 REM PROGRAMM-TYP SUCHEN                            <5CL>
1900 WINDOW 0,0,79,24                                   <O9E>
1910 PRINT "{CLR,DOWN}"                                 <GQF>
1920 PRINT TAB(26)"UCCCCCCCCCCCCCCCCCCCCCCCI"           <H4N>
1930 PRINT TAB(26)"B{2SPACE}PROGRAMMTYP SUCHEN{2S
     PACE}B"                                            <JHI>
1940 PRINT TAB(26)"JCCCCCCCCCCCCCCCCCCCCCCCK"           <I4L>
1950 PRINT "{DOWN,2SPACE}NAME{13SPACE}BLOECKE{6SP
     ACE}PRORAMMART{9SPACE}DISKETTENSEITE{5SPACE}
     FILETYP"                                           <6FR>
1960 FOR I=1 TO 80: PRINT "D";: NEXT I                  <MNI>
1970 PRINT "{DOWN,2SPACE}MOEGLICHE PROGRAMMTYPEN"       <232>
1980 PRINT "CCCCCCCCCCCCCCCCCCCCCCCCCCCC"               <DNC>
1990 FOR I=1 TO 7                                       <K21>
2000 PRINT TAB(8)"(";I;") ";TL$(I)                      <09J>
```

```
2010 NEXT I                                        <504>
2020 PRINT "{3DOWN}"                               <MLK>
2030 FOR I=1 TO 80: PRINT "B";: NEXT I             <DNI>
2040 WINDOW 0,23,79,24                             <3GJ>
2050 PRINT "{CLR} SUCH - TYP : ";CHR$(15);"B";CHR
     $(143);                                       <VOB>
2060 GET KEY E$: PRINT " "                         <8DM>
2070 IF VAL(E$)>7  OR VAL(E$)=0 THEN RETURN        <5GM>
2080 PRINT "{CLR}": IC=0                           <8S5>
2090 FOR I=1 TO PZ                                 <4M9>
2100 IF VAL(E$)=TY(I) THEN GOSUB 10250             <CRQ>
2110 NEXT I                                        <LQ4>
2120 WINDOW 0,23,79,24                             <2GN>
2130 PRINT "{CLR}";TAB(23)"KEINE WEITEREN EINTRAE
     GE VORHANDEN";                                <44P>
2140 GET KEY A$                                    <50D>
2150 RETURN                                        <027>
2160 REM DISKETTEN SUCHEN                          <144>
2170 PRINT "{CLR}"                                 <ALM>
2180 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCCI"    <1JD>
2190 PRINT TAB(25)"B{4SPACE}DISKETTEN SUCHEN{4SPA
     CE}B"                                         <900>
2200 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCCK"    <D3F>
2220 PRINT "{DOWN}SUCH DISKETTE ( 1 -";DZ;") : ";  <KIH>
2230 L=3: A1=48: A2=57: GOSUB 10100                <7JL>
2240 IF VAL(E$)=0 THEN RETURN                      <N4T>
2245 IF VAL(E$)>DZ OR (DB(2,1,VAL(E$))=0 AND DB(2
     ,2,VAL(E$))=0) THEN PRINT : PRINT "{DOWN}DIE
     SE DISKETTE UMFASST DIE DATEI NICHT !": GET
     KEY A$: RETURN                                <CGQ>
2250 PRINT : PRINT "{UP}DISKETTE : ";E$;" SEITE A
     {16SPACE}"                                    <TB1>
2260 PRINT TAB(40)"{UP}DISKETTE : ";E$;" SEITE B"  <JMQ>
2270 PRINT "{DOWN}NAME{13SPACE}PROGRAMMART{4SPACE
     }BLEOCKE NAME{13SPACE}PROGRAMMART{4SPACE}BLO
     ECKE"                                         <RGV>
2280 PRINT "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CC CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"       <I8T>
2290 WINDOW 0,21,79,21                             <HEO>
2300 PRINT "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CC CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC";      <2H6>
2310 WINDOW 0,9,40,20: PRINT "{CLR}";              <CIB>
2315 IF DB(2,1,VAL(E$))=1 THEN BEGIN               <DAV>
2320 FOR I=1 TO PZ                                 <9M9>
2330 IF DI(I)=2*VAL(E$)-1 THEN PRINT NA$(I);TAB(1
     7)AL$(AR(I));TAB(34)BL(I): IC=1               <3B1>
2340 NEXT I                                        <TR4>
2350 BEND : ELSE PRINT "KEINE EINTRAEGE VORHANDEN
     "                                             <TRP>
2360 WINDOW 40,9,79,20: PRINT "{CLR}";             <QLK>
2365 IF DB(2,2,VAL(E$))=1 THEN BEGIN               <NA7>
2370 FOR I=1 TO PZ                                 <CM9>
2380 IF DI(I)=2*VAL(E$) THEN PRINT NA$(I);TAB(17)
     AL$(AR(I));TAB(34)BL(I): IC=1                 <PO1>
2390 NEXT I                                        <LM4>
2400 BEND : ELSE PRINT "KEINE EINTRAEGE VORHANDEN
     "                                             <OBP>
2410 WINDOW 0,22,79,22                             <81T>
2420 PRINT "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CC CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"       <I4T>
2425 PRINT "FREIE BLOECKE :";: IF DB(2,1,VAL(E$))
     =1 THEN PRINT DB(1,1,VAL(E$));: ELSE PRINT "
      664";                                        <VBO>
2427 PRINT TAB(40)"FREIE BLOECKE :";: IF DB(2,2,V
     AL(E$))=1 THEN PRINT DB(1,2,VAL(E$));: ELSE
     PRINT " 664";                                 <4S3>
2440 WINDOW 0,24,79,24                             <PNR>
2450 PRINT "{CLR}";TAB(23)"KEINE WEITEREN EINTRAE
     GE VORHANDEN";                                <48P>
2460 GET KEY A$                                    <70D>
2470 RETURN                                        <OI7>
2480 REM FREIE BLOECKE SUCHEN                      <040>
2490 PRINT "{CLR,DOWN}"                            <VQF>
2500 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCI"     <1JD>
2510 PRINT TAB(25)"B{2SPACE}FREIE BLOECKE SUCHEN{
     2SPACE}B"                                     <O1G>
2520 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCK"     <E3F>
2530 PRINT "{DOWN,3SPACE}DISKETTENSEITE BLOECKE{3
     SPACE}DISKETTENSEITE BLOECKE{3SPACE}DISKETTE
     NSEITE BLOECKE "                              <BE3>
2540 PRINT "{2SPACE}CCCCCCCCCCCCCCCCCCCCRCCCC
     CCCCCCCCCCCCCCCCRCCCCCCCCCCCCCCCCCCCCCCC
     C"                                            <NGF>
2550 FOR I=1 TO 14                                 <N4V>
2560 PRINT "{26SPACE}B{24SPACE}B"                  <NM9>
2570 NEXT I                                        <5K4>
2580 PRINT "{2SPACE}CCCCCCCCCCCCCCCCCCCCECCCC
     CCCCCCCCCCCCCCCCECCCCCCCCCCCCCCCCCCCCCCC
     C"                                            <ER9>
2590 WINDOW 0,24,79,24                             <QFK>
2600 DO : PRINT "{CLR,2SPACE}MINDEST - SUCH - LAE
     NGE (BLOECKEN) : ";                           <O34>
2610 L=3: A1=48: A2=57: GOSUB 10100: LOOP WHILE V
     AL(E$)=0 OR VAL(E$)>664                       <M8N>
2620 IC=0: PRINT "{CLR}"                           <RL4>
2630 FOR I=1 TO DZ                                 <7MT>
2635 FOR A=1 TO 2                                  <PQ3>
2637 FB=DB(1,A,I)                                  <9BO>
2638 IF DB(2,A,I)=0 THEN FB=664                    <OA8>
2640 IF FB>=VAL(E$) THEN BEGIN                     <PH1>
2650 IC=IC+1: IF IC=40 THEN BEGIN                  <TJM>
2660 IC=1: WINDOW 0,24,79,24: PRINT "{CLR}";TAB(2
     3)"WEITERE EINTRAEGE VORHANDEN";              <JNU>
2670 GET KEY A$: PRINT "{CLR}"                     <00V>
2680 BEND                                          <9TE>
2690 IF IC=1 THEN WINDOW 2,8,25,21: PRINT "{CLR}"
     ;                                             <L01>
2700 IF IC=14 THEN WINDOW 27,8,50,21: PRINT "{CLR
     }";                                           <9BJ>
2710 IF IC=27 THEN WINDOW 52,8,75,21: PRINT "{CLR
     }";                                           <8U5>
2720 D$=STR$(I): IF A=1 THEN S$="A": ELSE S$="B"   <ML3>
2730 DO WHILE LEN(D$)<3: D$=" "+D$: LOOP           <MPH>
2740 PRINT "{4SPACE}(";D$;" ";S$;"){5SPACE}";FB    <6UO>
2750 GET A$: IF A$<>"" THEN GET KEY A$             <LAE>
2760 BEND                                          <PRE>
2770 NEXT A,I                                      <G4A>
2780 WINDOW 0,24,79,24                             <R7H>
2790 PRINT "{CLR}";TAB(23)"KEINE WEITEREN EINTRAE
     GE VORHANDEN";                                <K2P>
2800 GET KEY A$                                    <0GD>
2810 RETURN                                        <PE7>
2820 REM DATEN DRUCKEN                             <4KI>
2825 IF PZ=0 THEN RETURN                           <84A>
2830 IF DA$="" THEN BEGIN                          <BLQ>
2840 PRINT "{CLR}"                                 <ETM>
2850 PRINT "BITTE GEBEN SIE DAS HEUTIGE DATUM EIN
     "                                             <DIC>
2860 FOR I=1 TO 80: PRINT "C";: NEXT I             <OBI>
2870 PRINT "{DOWN}TAG (1-31){5SPACE}: ";           <7I4>
2880 L=2: A1=48: A2=57: GOSUB 10100: TA$=E$        <97U>
2890 PRINT : PRINT "{DOWN}MONAT (1-12){3SPACE}: "
     ;                                             <S98>
2900 L=2: A1=48: A2=57: GOSUB 10100: MO$=E$        <K8A>
2910 PRINT : PRINT "{DOWN}JAHR (Z.B.:87) : ";      <BKO>
2920 L=2: A1=48: A2=57: GOSUB 10100: DA$=TA$+"."+
     MO$+".19"+E$                                  <MSQ>
2930 PRINT : PRINT "{DOWN}HEUTIGES DATUM : ";DA$   <O14>
2940 PRINT "{DOWN}EINGABE IN ORDNUNG ?"            <D94>
2950 DO : GET KEY A$: LOOP WHILE A$<>"J" AND A$<>
     "N"                                           <05B>
2960 IF A$="N" THEN GOTO 2840                      <V9O>
2970 BEND                                          <DLU>
2980 WINDOW 0,0,79,24                              <OPC>
2990 PRINT "{CLR,DOWN}"                            <IQF>
3000 PRINT TAB(26)"UCCCCCCCCCCCCCCCCCCCCCCI"       <GKN>
3010 PRINT TAB(26)"B{4SPACE}DATEN DRUCKEN{5SPACE}
     B"                                            <PBA>
3020 PRINT TAB(26)"JCCCCCCCCCCCCCCCCCCCCCCK"       <JSL>
3030 PRINT TAB(27)"{DOWN}A : LISTE ALPHABETISCH"   <999>
3040 PRINT TAB(27)"{DOWN}B : LISTE DISKETTEN"      <G87>
3050 PRINT TAB(27)"{DOWN}C : LISTE PROGRAMMART"    <LHK>
3060 PRINT TAB(27)"{DOWN}D : LISTE PROGRAMMTYP"    <K1I>
3070 PRINT TAB(27)"{DOWN}E : LISTE FREIE BLOECKE"  <ITI>
3080 PRINT TAB(27)"{DOWN}F : HAUPTMENUE"           <QBT>
3090 PRINT TAB(29)"{2DOWN}TASTE DRUECKEN"          <D9H>
3100 DO                                            <UFR>
3110 GET KEY A$                                    <58D>
3120 IF ASC(A$)<71 AND ASC(A$)>64 THEN EXIT        <Q75>
3130 LOOP                                          <44P>
3140 IF A$="F" THEN RETURN                         <GJI>
3150 ON  ASC(A$)- 64 GOSUB 3170,3640,4400,5120,57
     30                                            <MS6>
3160 GOTO 2980                                     <782>
3170 REM LISTE ALPHABETISCH                        <0C1>
3180 WINDOW 0,0,79,24                              <Q96>
3190 PRINT "{CLR,DOWN}"                            <OQF>
3200 PRINT TAB(26)"UCCCCCCCCCCCCCCCCCCCCCCCCI"     <F3D>
3210 PRINT TAB(26)"B{3SPACE}LISTE ALPHABETISCH{3S
     PACE}B"                                       <OE8>
3220 PRINT TAB(26)"JCCCCCCCCCCCCCCCCCCCCCCCCK"     <0JF>
3230 WINDOW 0,10,79,24                             <ADK>
3240 PRINT TAB(18)"VON NAME ( ";NA$(1);" - ";NA$(
     PZ);" ) : ";                                  <BH0>
3250 L=16: A1=32: A2=95: GOSUB 10100: G1$=E$: PRI
     NT                                            <0N8>
3260 IF G1$<LEFT$(NA$(1),LEN(G1$)) OR G1$>LEFT$(N
     A$(PZ),LEN(G1$)) THEN GOTO 3170               <S93>
3270 PRINT TAB(18)"{DOWN}BIS NAME ( ";G1$;" - ";N
     A$(PZ);" ) : ";                               <O5M>
3280 L=16: A1=32: A2=95: GOSUB 10100: G2$=E$: PRI
     NT                                            <2F8>
3290 IF G2$<LEFT$(G1$,LEN(G2$)) OR G2$>LEFT$(NA$(
     PZ),LEN(G2$)) THEN GOTO 3170                  <T7G>
3300 PRINT TAB(18)"{DOWN}BENUTZEN SIE (E)INZEL OD
     ER (R)OLLENPAPIER ?"                          <GB2>
3310 DO : GET KEY DM$: LOOP WHILE DM$<>"E" AND DM
     $<>"R"                                        <278>
3320 PRINT TAB(18)"{DOWN}DRUCKER O.K. DANN {RVSON
     }RETURN{RVOFF} DRUECKEN"                      <EST>
3330 DO : GET KEY A$: LOOP WHILE ASC(A$)<>13       <KTN>
3340 OPEN 1,4,SA: SE=1: PRINT "{CLR}"              <RJK>
3350 PRINT#1,"{9SPACE}***********************
     ****************************************
     "                                             <2GC>
3360 PRINT#1,"{33SPACE}DISK-MANAGER VERSION 1.0"   <219>
3370 PRINT#1,"{9SPACE}***********************
     ****************************************
     ": Z=5: I=0                                   <ON9>
3380 DO                                            <IEB>
3390 PRINT#1                                       <2M3>
3400 PRINT#1,"{9SPACE}";LEFT$("DATEI - NAME : "+D
     N$+"{16SPACE}",33);LEFT$("- SEITE : "+RIGHT$
```

Listing 2. (Fortsetzung)

```
     (STR$(SE),LEN(STR$(SE))-1)+" -{15SPACE}",27)
     ;DA$                                              <TTK>
3420 PRINT#1                                           <223>
3430 PRINT#1,"{9SPACE}NR.{2SPACE}NAME{13SPACE}BLO
     ECKE PRORAMMART{6SPACE}DISKETTENSEITE FILETY
     P"                                                <3BU>
3440 PRINT#1,"{9SPACE}---------------------------
     ---------------------------------------------
     "                                                 <IJH>
3460 DO : I=I+1                                        <JIT>
3470 PRINT TAB(30)"NAME : ";NA$(I);"{16SPACE,UP}"      <OAH>
3480 IF LEFT$(NA$(I),LEN(G1$))>=G1$ AND LEFT$(NA$
     (I),LEN(G2$))<=G2$ THEN BEGIN                     <QFD>
3490 Z=Z+1: GOSUB 10250                                <GIH>
3500 PRINT#1,"{9SPACE}";LEFT$(RIGHT$(STR$(Z-3),LE
     N(STR$(Z-3))-1)+"{4SPACE}",5);LEFT$(NA$+"{18
     SPACE}",19);                                      <L7S>
3505 PRINT#1,LEFT$(RIGHT$(STR$(BL),LEN(STR$(BL))-
     1)+"{5SPACE}",6);LEFT$(AL$(AR(I))+"{17SPACE}
     ",18);LEFT$("("+D$+"){14SPACE}",17);TL$(TY(I
     ))                                                <2I2>
3510 BEND                                              <5OU>
3520 IF INT(Z/60)=Z/60 OR I=PZ THEN EXIT               <JES>
3530 LOOP                                              <40P>
3540 SE=SE+1: IF I=PZ THEN EXIT                        <J3H>
3550 IF DM$="E" THEN BEGIN                             <P7P>
3560 PRINT#1,CHR$(12)                                  <EDL>
3570 PRINT "{CLR}";TAB(18)"BITTE NEUES BLATT EINS
     PANNEN UND {RVSON}RETURN{RVOFF} DRUECKEN"         <9DH>
3580 GET KEY A$: PRINT "{CLR}"                         <6G5>
3590 BEND : ELSE FOR A=1 TO 4: PRINT#1,CHR$(10):
     NEXT A                                            <QRG>
3600 LOOP                                              <G29>
3610 PRINT#1,CHR$(12)                                  <E5K>
3620 CLOSE 1: CLOSE 2: CLOSE 3: DM$=""                 <U1G>
3630 RETURN                                            <OQ7>
3640 REM LISTE DISKETTEN                               <34C>
3650 WINDOW 0,0,79,24                                  <Q57>
3660 PRINT "{CLR,DOWN}"                                <OAF>
3670 PRINT TAB(26)"UCCCCCCCCCCCCCCCCCCCCCCCCI"         <RM1>
3680 PRINT TAB(26)"B{4SPACE}LISTE DISKETTEN{4SPAC
     E}B"                                              <CSR>
3690 PRINT TAB(26)"JCCCCCCCCCCCCCCCCCCCCCCCCK"         <T63>
3700 WINDOW 0,10,79,24                                 <85T>
3720 PRINT TAB(18)"VON DISKETTE ( 1 -";DZ;") : ";      <G45>
3730 L=3: A1=48: A2=57: GOSUB 10100: DL=VAL(E$):
     PRINT                                             <3AG>
3740 IF DL=0 OR DL>DZ THEN GOTO 3650                   <I41>
3750 PRINT TAB(18)"{DOWN}BIS DISKETTE (";DL;"-";D
     Z;") : ";                                         <CVF>
3760 L=3: A1=48: A2=57: GOSUB 10100: DH=VAL(E$):
     PRINT                                             <9EG>
3770 IF DH<DL OR DH>DZ THEN GOTO 3650                  <41R>
3780 PRINT TAB(18)"{DOWN}BENUTZEN SIE (E)INZEL OD
     ER (R)OLLENPAPIER ?"                              <GB2>
3790 DO : GET KEY DM$: LOOP WHILE DM$<>"E" AND DM
     $<>"R"                                            <278>
3800 PRINT TAB(18)"{DOWN}DRUCKER O.K. DANN {RVSON
     }RETURN{RVOFF} DRUECKEN"                          <QUD>
3810 DO : GET KEY A$: LOOP WHILE ASC(A$)<>13           <HT3>
3820 OPEN 1,4,SA: SE=1: ZE=3: PRINT "{CLR}"            <5NN>
3830 PRINT#1,"{9SPACE}***************************
     *********************************************
     "                                                 <IGC>
3840 PRINT#1,"{33SPACE}DISK-MANAGER VERSION 1.0"       <A49>
3850 PRINT#1,"{9SPACE}***************************
     *********************************************
     "                                                 <KGC>
3860 DO                                                <ADB>
3870 PRINT#1                                           <1M3>
3880 PRINT#1,"{9SPACE}";LEFT$("DATEI - NAME : "+D
     N$+"{16SPACE}",33);LEFT$("- SEITE : "+RIGHT$
     (STR$(SE),LEN(STR$(SE))-1)+" -{15SPACE}",27)
     ;DA$                                              <STK>
3900 DO                                                <EDR>
3910 A=0: B=0: PRINT "{CLR}"                           <GTL>
3920 FOR I=1 TO PZ                                     <PM9>
3930 IF DI(I)=2*DL-1 THEN A=A+1: A(A)=I: PRINT TA
     B(20)"DISKETTE : ";DL;" SEITE A :";NA$(I)         <E0D>
3940 IF DI(I)=2*DL   THEN B=B+1: B(B)=I: PRINT TA
     B(20)"DISKETTE : ";DL;" SEITE B :";NA$(I)         <2PN>
3950 NEXT I                                            <LA4>
3960 IF ZE+A+B+16>63 THEN EXIT                         <95F>
3970 ZE=ZE+A+B+16                                      <PKK>
3980 PRINT#1: PRINT#1                                  <GC0>
3990 PRINT#1,"{9SPACE}DISKETTE :";DL;" SEITE : A"      <42B>
4000 PRINT#1                                           <0Q3>
4010 PRINT#1,"{9SPACE}NR.{5SPACE}NAME{16SPACE}BLO
     ECKE{5SPACE}PRORAMMART{11SPACE}FILETYP"           <FBU>
4020 PRINT#1,"{9SPACE}---------------------------
     ---------------------------------------------"    <2RL>
4030 Z=1                                               <JE9>
4040 IF DB(2,1,DL)=1 THEN BEGIN                        <JTN>
4050 FOR I=1 TO A                                      <VRQ>
4060 PRINT#1,"{9SPACE}";LEFT$(RIGHT$(STR$(Z),LEN(
     STR$(Z))-1)+"{7SPACE}",8);LEFT$(NA$(A(I))+"{
     20SPACE}",21);                                    <4NL>
4065 BL = BL(A(I)): PRINT#1,LEFT$(RIGHT$(STR$(BL)
     ,LEN(STR$(BL))-1)+"{10SPACE}",11);LEFT$(AL$(
     AR(A(I)))+"{20SPACE}",23);TL$(TY(A(I))): Z=Z
     +1                                                <J6E>
4070 NEXT I                                            <L64>
4075 BEND : ELSE PRINT#1,"{9SPACE}KEINE EINTRAEGE
      VORHANDEN"                                       <544>
4080 PRINT#1,"{9SPACE}---------------------------
     ---------------------------------------------"    <3JL>
4090 PRINT#1,"{9SPACE}FREIE BLOECKE : ";: IF DB(2
     ,1,DL)=1 THEN PRINT#1,DB(1,1,DL): ELSE PRINT
     #1," 664"                                         <TJ4>
4100 PRINT#1: PRINT#1                                  <GS2>
4110 PRINT#1,"{9SPACE}DISKETTE :";DL;" SEITE : B"      <4UB>
4120 PRINT#1                                           <0Q3>
4130 PRINT#1,"{9SPACE}NR.{5SPACE}NAME{16SPACE}BLO
     ECKE{5SPACE}PRORAMMART{11SPACE}FILETYP"           <FBU>
4140 PRINT#1,"{9SPACE}---------------------------
     ---------------------------------------------"    <2BL>
4150 Z=1                                               <RF9>
4160 IF DB(2,2,DL)=1 THEN BEGIN                        <3PN>
4170 FOR I=1 TO B                                      <VNR>
4180 PRINT#1,"{9SPACE}";LEFT$(RIGHT$(STR$(Z),LEN(
     STR$(Z))-1)+"{7SPACE}",8);LEFT$(NA$(B(I))+"{
     20SPACE}",21);                                    <2RC>
4185 BL = BL(B(I)): PRINT#1,LEFT$(RIGHT$(STR$(BL)
     ,LEN(STR$(BL))-1)+"{10SPACE}",11);LEFT$(AL$(
     AR(B(I)))+"{20SPACE}",23);TL$(TY(B(I))): Z=Z
     +1                                                <6OL>
4190 NEXT I                                            <LA4>
4195 BEND : ELSE PRINT#1,"{9SPACE}KEINE EINTRAEGE
      VORHANDEN"                                       <584>
4200 PRINT#1,"{9SPACE}---------------------------
     ---------------------------------------------"    <23L>
4210 PRINT#1,"{9SPACE}FREIE BLOECKE : ";: IF DB(2
     ,2,DL)=1 THEN PRINT#1,DB(1,2,DL): ELSE PRINT
     #1," 664"                                         <DL4>
4220 DO : DL=DL+1                                      <24H>
4222 IF DB(2,1,DL)=1 OR DB(2,2,DL)=1 OR DL>DH THE
     N EXIT                                            <DHC>
4224 LOOP                                              <089>
4230 IF DL>DH THEN EXIT                                <S96>
4240 LOOP                                              <GA9>
4250 IF DL>DH THEN EXIT                                <SH5>
4260 IF DM$="E" THEN BEGIN                             <85T>
4270 PRINT#1,CHR$(12)                                  <AL6>
4280 PRINT "{CLR}";TAB(18)"BITTE NEUES BLATT EINS
     PANNEN UND {RVSON}RETURN{RVOFF} DRUECKEN"         <T31>
4290 DO : GET KEY A$: LOOP UNTIL A$=CHR$(13)           <9SB>
4300 BEND : ELSE BEGIN                                 <IJK>
4310 DO WHILE ZE<>63: PRINT#1: ZE=ZE+1: LOOP           <40L>
4320 FOR I=1 TO 7: PRINT#1: NEXT I                     <P8K>
4330 BEND                                              <DHU>
4340 ZE=0: SE=SE+1                                     <GH5>
4350 LOOP                                              <SBP>
4360 PRINT#1,CHR$(12)                                  <AD5>
4370 CLOSE 1: CLOSE 2: CLOSE 3                         <853>
4380 DM$=""                                            <KJH>
4390 RETURN                                            <P27>
4400 REM LISTE PROGRAMMART                             <44G>
4410 PRINT "{CLR}"                                     <FLM>
4420 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCI"         <U21>
4430 PRINT TAB(25)"B{3SPACE}LISTE PROGRAMMART{3SP
     ACE}B"                                            <AR8>
4440 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCK"         <PI3>
4450 WINDOW 0,5,79,24                                  <H0B>
4460 PRINT "{24SPACE}PROGRAMMARTEN"                    <OH8>
4465 PRINT "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"                  <OTO>
4470 FOR I=1 TO 6                                      <K61>
4480 PRINT " (";I;") ";TAB(8)AL$(I);TAB(24)" (";I
     +6;") ";TAB(32)AL$(I+6);TAB(48)" (";I+12;")
     ";TAB(56)AL$(I+12)                                <C1V>
4490 NEXT I                                            <504>
4495 PRINT TAB(48)" ( 19 ) ";AL$(19)                   <TJ3>
4500 PRINT TAB(18)"{DOWN}VON ART ({2SPACE}1{2SPAC
     E}- 19 ) : ";                                     <BCE>
4510 L=2: A1=48: A2=57: GOSUB 10100: G1=VAL(E$):
     PRINT                                             <CIV>
4520 IF G1=0 OR G1>19 THEN GOTO 4400                   <J5M>
4530 PRINT TAB(18)"{DOWN}BIS ART ( ";G1;" -{2SPAC
     E}19 ) : ";                                       <VB6>
4540 L=2: A1=48: A2=57: GOSUB 10100: G2=VAL(E$):
     PRINT                                             <UIV>
4550 IF G2<G1 OR G2>19 THEN GOTO 4400                  <L2H>
4560 PRINT TAB(18)"{DOWN}BENUTZEN SIE (E)INZEL OD
     ER (R)OLLENPAPIER ?"                              <6B2>
4570 DO : GET KEY DM$: LOOP WHILE DM$<>"E" AND DM
     $<>"R"                                            <978>
4580 PRINT TAB(18)"{DOWN}DRUCKER O.K. DANN {RVSON
     }RETURN{RVOFF} DRUECKEN"                          <KVL>
4590 DO : GET KEY A$: LOOP WHILE ASC(A$)<>13           <IDD>
4600 OPEN 1,4,SA: SE=1: PRINT "{CLR}"                  <V3K>
4610 PRINT#1,"{9SPACE}***************************
     *********************************************
     "                                                 <GGC>
4620 PRINT#1,"{33SPACE}DISK-MANAGER VERSION 1.0"       <65P>
4630 PRINT#1,"{9SPACE}***************************
     *********************************************
     ": Z=5                                            <NPN>
4640 PRINT#1                                           <1Q3>
4650 PRINT#1,"{9SPACE}";LEFT$("DATEI - NAME : "+D
     N$+"{16SPACE}",33);LEFT$("- SEITE : "+RIGHT$
     (STR$(SE),LEN(STR$(SE))-1)+" -{15SPACE}",27)
     ;DA$                                              <C7K>
4680 DO : I=0                                          <HJ7>
4690 DO : I=I+1                                        <MAT>
4700 IF AR(I)=G1 OR I=PZ THEN EXIT                     <G3I>
4710 LOOP                                              <CDP>
4720 IF AR(I)=G1 THEN BEGIN                            <H34>
4730 IF Z+5>=61 THEN GOSUB 4930                        <NSK>
```

```
4740 GOSUB 5040                                          <07P>
4750 I=0: P=0: S=1                                       <5HV>
4760 DO : I=I+1                                          <L2T>
4770 PRINT TAB(30)"NAME : ";NA$(I);"{16SPACE,UP}"        <UQB>
4780 IF AR(I)=G1 THEN BEGIN                              <51K>
4790 IF S=0 THEN S=1: GOSUB 5040                         <GD8>
4800 P=P+1: Z=Z+1: GOSUB 10250                           <CG6>
4810 PRINT#1,"{9SPACE}";LEFT$(RIGHT$(STR$(P),LEN(
     STR$(P))-1)+"{4SPACE}",5);LEFT$(NA$+"{18SPAC
     E}",19);                                            <R92>
4815 PRINT#1,LEFT$(RIGHT$(STR$(BL),LEN(STR$(BL))-
     1)+"{6SPACE}",6);LEFT$(AL$(AR(I))+"{17SPACE}
     ",18);LEFT$("("+D$+") {14SPACE}",17);TL$(TY(I
     ))                                                  <N36>
4820 BEND                                                <HQE>
4830 IF I=PZ THEN EXIT                                   <6M2>
4840 IF Z>=58 THEN GOSUB 4930                            <NED>
4850 LOOP                                                <40P>
4860 BEND                                                <1OE>
4870 G1=G1+1                                             <6AL>
4880 IF G1>G2 THEN EXIT                                  <B98>
4890 LOOP                                                <K6P>
4900 PRINT#1,CHR$(12): DM$=""                            <R9N>
4910 CLOSE 1: CLOSE 2: CLOSE 3                           <0C3>
4920 RETURN                                              <8S7>
4930 SE=SE+1                                             <PPD>
4940 IF DM$="E" THEN BEGIN                               <S7D>
4950 PRINT#1,CHR$(12)                                    <FLI>
4960 PRINT "{CLR}";TAB(18)"BITTE NEUES BLATT EINS
     PANNEN UND {RVSON}RETURN{RVOFF} DRUECKEN"           <TF1>
4970 GET KEY A$: PRINT "{CLR}"                           <30J>
4980 BEND : ELSE DO : PRINT#1: Z=Z+1: LOOP UNTIL
     Z=66                                                <7MA>
4990 Z=0: S=0                                            <CVR>
5000 BEND                                                <9HE>
5010 PRINT#1,"{9SPACE}";LEFT$("DATEI - NAME : "+D
     N$+"{16SPACE}",33);LEFT$("- SEITE : "+RIGHT$
     (STR$(SE),LEN(STR$(SE))-1)+" -{15SPACE}",27)
     ;DA$                                                <CVK>
5025 Z=3                                                 <S7K>
5030 RETURN                                              <PI7>
5040 PRINT#1                                             <0Q3>
5050 PRINT#1,"{36SPACE}";AL$(G1)                         <7HC>
5060 PRINT#1: PRINT "{CLR}";TAB(30)AL$(G1): PRINT
      TAB(29)"CCCCCCCCCCCCCCCCC": PRINT                  <PCT>
5070 PRINT#1,"{9SPACE}NR.{2SPACE}NAME{13SPACE}BLO
     ECKE PRORAMMART{8SPACE}DISKETTENSEITE FILETY
     P"                                                  <9BU>
5080 PRINT#1,"{9SPACE}-----------------------------
     ----------------------------------------
     "                                                   <EJH>
5100 Z=Z+4                                               <S31>
5110 RETURN                                              <PI7>
5120 REM LISTE PROGRAMMTYP                               <74S>
5130 PRINT "{CLR}"                                       <CLM>
5140 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCI"           <V21>
5150 PRINT TAB(25)"B{3SPACE}LISTE PROGRAMMTYP{3SP
     ACE}B"                                              <IO8>
5160 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCK"           <O23>
5170 WINDOW 0,5,79,24                                    <GGD>
5180 PRINT "{4SPACE}TYPEN"                               <70I>
5185 PRINT "CCCCCCCCCCCC"                                <OSB>
5190 FOR I=1 TO 7                                        <K21>
5200 PRINT " (";I;") ";TAB(8)TL$(I)                      <H25>
5210 NEXT I                                              <504>
5220 PRINT TAB(18)"{DOWN}VON TYP ({2SPACE}1{2SPAC
     E}- 7 ) : ";CHR$(15);"@{LEFT}";CHR$(143);           <O7Q>
5230 GET KEY G1: PRINT G1                                <813>
5240 IF G1=0 OR G1>7  THEN RETURN                        <G30>
5250 PRINT TAB(18)"{DOWN}BIS TYP ( ";G1;" -{2SPAC
     E}7 ) : ";CHR$(15);"@{LEFT}";CHR$(143);             <7MM>
5260 GET KEY G2: PRINT G2                                <997>
5270 IF G2<G1 OR G2>7 THEN RETURN                        <L0C>
5280 PRINT TAB(18)"{DOWN}BENUTZEN SIE (E)INZEL OD
     ER (R)OLLENPAPIER ?"                                <2B2>
5290 DO : GET KEY DM$: LOOP WHILE DM$<>"E" AND DM
     $<>"R"                                              <B78>
5300 PRINT TAB(18)"{DOWN}DRUCKER O.K. DANN {RVSON
     }RETURN{RVOFF} DRUECKEN"                            <KVL>
5310 DO : GET KEY A$: LOOP WHILE ASC(A$)<>13             <IDD>
5320 OPEN 1,4,SA: SE=1: PRINT "{CLR}"                    <S3K>
5330 PRINT#1,"{9SPACE}******************************
     ****************************************
     "                                                   <SGC>
5340 PRINT#1,"{33SPACE}DISK-MANAGER VERSION 1.0"         <M7P>
5350 PRINT#1,"{9SPACE}******************************
     ****************************************
     ": Z=5                                              <N9L>
5360 PRINT#1                                             <0A3>
5370 PRINT#1,"{9SPACE}";LEFT$("DATEI - NAME : "+D
     N$+"{16SPACE}",33);LEFT$("- SEITE : "+RIGHT$
     (STR$(SE),LEN(STR$(SE))-1)+" -{15SPACE}",27)
     ;DA$                                                <CVK>
5400 DO : I=0                                            <H37>
5410 DO : I=I+1                                          <KAT>
5420 IF TY(I)=G1 OR I=PZ THEN EXIT                       <IBQ>
5430 LOOP                                                <CDP>
5440 IF TY(I)=G1 THEN BEGIN                              <B2C>
5450 IF Z+5>=61 THEN Z=Z-6: GOSUB 4930                   <VKU>
5460 GOSUB 5650                                          <RFK>
5470 I=0: P=0: S=1                                       <5HV>
5480 DO : I=I+1                                          <M2T>
5490 PRINT TAB(30)"NAME : ";NA$(I);"{16SPACE,UP}"        <TQ7>
5500 IF TY(I)=G1 THEN BEGIN                              <73S>
5510 IF S=0 THEN S=1: GOSUB 5650                         <JL5>
5520 P=P+1: Z=Z+1: GOSUB 10250                           <D86>
5530 PRINT#1,"{9SPACE}";LEFT$(RIGHT$(STR$(P),LEN(
     STR$(P))-1)+"{4SPACE}",5);LEFT$(NA$+"{18SPAC
     E}",19);                                            <QH2>
5535 PRINT#1,LEFT$(RIGHT$(STR$(BL),LEN(STR$(BL))-
     1)+"{5SPACE}",6);LEFT$(AL$(AR(I))+"{17SPACE}
     ",18);LEFT$("("+D$+") {14SPACE}",17);TL$(TY(I
     ))                                                  <4AP>
5540 BEND                                                <HME>
5550 IF I=PZ THEN EXIT                                   <764>
5560 IF Z>=61 THEN Z=Z-6: GOSUB 4930                     <SIK>
5570 LOOP                                                <KEP>
5580 BEND                                                <1GE>
5590 G1=G1+1                                             <4AL>
5600 IF G1>G2 THEN EXIT                                  <990>
5610 LOOP                                                <KEP>
5620 PRINT#1,CHR$(12): DM$=""                            <O9N>
5630 CLOSE 1: CLOSE 2: CLOSE 3                           <003>
5640 RETURN                                              <947>
5650 PRINT#1                                             <1E3>
5660 PRINT#1,"{38SPACE}";TL$(G1)                         <IHL>
5670 PRINT#1: PRINT "{CLR}";TAB(38)TL$(G1): PRINT
      TAB(37)"CCCCC": PRINT                              <S4U>
5680 PRINT#1,"{9SPACE}NR.{2SPACE}NAME{13SPACE}BLO
     ECKE PRORAMMART{8SPACE}DISKETTENSEITE FILETY
     P"                                                  <8RU>
5690 PRINT#1,"{9SPACE}-----------------------------
     ----------------------------------------
     "                                                   <3JH>
5710 Z=Z+4                                               <KE1>
5720 RETURN                                              <9C7>
5730 REM LISTE FREIE BLOECKE                             <4CH>
5740 PRINT "{CLR}"                                       <EDM>
5750 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCI"           <UM1>
5760 PRINT TAB(25)"B{2SPACE}LISTE FREIE BLOECKE{2
     SPACE}B"                                            <QFR>
5770 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCK"           <PM3>
5780 WINDOW 0,5,79,24                                    <GSC>
5790 PRINT TAB(18)"{DOWN}VON DISKETTE ({2SPACE}1{
     2SPACE}- ";DZ;" ) : ";                              <819>
5800 L=3: A1=48: A2=57: GOSUB 10100: G1=VAL(E$):
     PRINT                                               <IIV>
5810 IF G1=0 OR G1>D  THEN RETURN                        <OL0>
5820 PRINT TAB(18)"{DOWN}BIS DISKETTE ( ";G1;" -
     ";DZ;" ) : ";                                       <CO3>
5830 L=3: A1=48: A2=57: GOSUB 10100: G2=VAL(E$):
     PRINT                                               <CIV>
5840 IF G2<G1 OR G2>DZ THEN RETURN                       <V8G>
5850 PRINT TAB(18)"{DOWN}BENUTZEN SIE (E)INZEL OD
     ER (R)OLLENPAPIER ?"                                <RB2>
5860 DO : GET KEY DM$: LOOP WHILE DM$<>"E" AND DM
     $<>"R"                                              <6N8>
5870 PRINT TAB(18)"{DOWN}DRUCKER O.K. DANN {RVSON
     }RETURN{RVOFF} DRUECKEN"                            <5TH>
5880 DO : GET KEY A$: LOOP WHILE ASC(A$)<>13             <LLI>
5890 OPEN 1,4,SA: SE=1: PRINT "{CLR}"                    <RBK>
5900 PRINT#1,"{9SPACE}******************************
     ****************************************
     "                                                   <BGC>
5910 PRINT#1,"{33SPACE}DISK-MANAGER VERSION 1.0"         <K3H>
5920 PRINT#1,"{9SPACE}******************************
     ****************************************
     ": Z=3: I=G1-1                                      <FF7>
5930 DO                                                  <LEN>
5940 PRINT#1                                             <2I3>
5950 PRINT#1,"{9SPACE}";LEFT$("DATEI - NAME : "+D
     N$+"{16SPACE}",33);LEFT$("- SEITE : "+RIGHT$
     (STR$(SE),LEN(STR$(SE))-1)+" -{15SPACE}",27)
     ;DA$                                                <DJK>
5970 PRINT#1                                             <2U3>
5980 PRINT#1,"{9SPACE}DISKETTE{3SPACE}!{3SPACE}FR
     EIE BLOECKE SEITE A{3SPACE}!{3SPACE}FREIE BL
     OECKE SEITE B"                                      <DLD>
5990 PRINT#1,"{9SPACE}-----------------------------
     ---------------------------------------"            <D3N>
6010 DO                                                  <9D7>
6020 I=I+1: Z=Z+2                                        <S36>
6025 IF I>G2 OR Z>=60 THEN EXIT                          <75A>
6030 PRINT TAB(20)"DISKETTE : ";I;"{UP}"                 <JNF>
6039 TRAP                                                <RF5>
6040 PRINT#1,"{12SPACE}";LEFT$(RIGHT$(STR$(I),LEN
     (STR$(I))-1)+"{7SPACE}",8);"!{10SPACE}";            <P2Q>
6042 IF DB(2,1,I)=1 THEN PRINT#1,LEFT$(RIGHT$(STR
     $(DB(1,1,I)),LEN(STR$(DB(1,1,I)))-1)+"{16SPA
     CE}",17);: ELSE PRINT#1," 664{14SPACE}";            <DHN>
6045 PRINT#1,"!{10SPACE}";: IF DB(2,2,I)=1 THEN P
     RINT#1,DB(1,2,I): ELSE PRINT#1," 664"               <EC1>
6050 LOOP                                                <4CP>
6060 SE=SE+1: Z=0: IF I>G2 THEN EXIT                     <ODL>
6070 IF DM$="E" THEN BEGIN                               <549>
6080 PRINT#1,CHR$(12)                                    <AD5>
6090 PRINT "{CLR}";TAB(18)"BITTE NEUES BLATT EINS
     PANNEN UND {RVSON}RETURN{RVOFF} DRUECKEN"           <95H>
6100 GET KEY A$: PRINT "{CLR}"                           <CHD>
6110 BEND : ELSE FOR A=1 TO 7: PRINT#1,CHR$(10):
     NEXT A                                              <0TG>
6120 LOOP                                                <GE9>
6130 PRINT#1,CHR$(12)                                    <85C>
```

Listing 2. (Fortsetzung)

```
6140 CLOSE 1: CLOSE 2: CLOSE 3: DM$=""                      <65G>
6150 RETURN                                                  <P27>
6160 REM DATEN EDITIEREN                                     <54K>
6170 SI=0                                                    <8TB>
6180 WINDOW 0,0,79,24                                        <0PC>
6190 PRINT "{CLR}";                                          <T7G>
6200 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCI"               <63D>
6210 PRINT TAB(25)"B{5SPACE}DATEN EDITIEREN{4SPAC
     E}B"                                                    <DED>
6220 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCK"               <9JF>
6230 PRINT TAB(27)"{DOWN}A : NACHLADE-PRG'S MODUS
      AN ";: IF TR=1 THEN PRINT "{3LEFT}AUS": ELS
     E PRINT "  "                                            <HDS>
6240 PRINT TAB(27)"{DOWN}B : DATEN LESEN"                    <QFG>
6250 PRINT TAB(27)"{DOWN}C : DATEN AENDERN"                  <BMB>
6260 PRINT TAB(27)"{DOWN}D : DATEN EINFUEGEN"                <P7G>
6270 PRINT TAB(27)"{DOWN}E : PROGRAMMARTEN AENDER
     N"                                                      <JHK>
6280 PRINT TAB(27)"{DOWN}F : PROGRAMMTYPEN AENDER
     N"                                                      <MPL>
6290 PRINT TAB(27)"{DOWN}G : DIRECTORY"                      <G00>
6300 PRINT TAB(27)"{DOWN}H : DISKETTENKOMMANDO"              <IQN>
6310 PRINT TAB(27)"{DOWN}I : HAUPTMENUE"                     <QVV>
6320 PRINT TAB(29)"{2DOWN}TASTE DRUECKEN"                    <SPN>
6330 DO                                                      <DDN>
6340 GET KEY A$                                              <00D>
6350 IF ASC(A$)<74 AND ASC(A$)>64 THEN EXIT                  <5KR>
6360 LOOP                                                    <GE9>
6370 IF A$="I" THEN GOTO 8560                                <RAO>
6380 ON  ASC(A$)- 64 GOSUB 6400,6430,7050,8080,81
     70,8360,9840,9730                                       <VKM>
6390 GOTO 6180                                               <0G2>
6400 REM NACHLADE PRG-MODUS                                  <640>
6410 IF TR=0 THEN TR=1: ELSE TR=0                            <VMS>
6420 RETURN                                                  <907>
6430 REM DATEN LESEN                                         <6SR>
6440 PRINT "{CLR}": SI=2                                     <N3Q>
6450 PRINT TAB(27)"UCCCCCCCCCCCCCCCCCCCI"                    <09R>
6460 PRINT TAB(27)"B{4SPACE}DATEN LESEN{4SPACE}B"            <PAK>
6470 PRINT TAB(27)"JCCCCCCCCCCCCCCCCCCCK"                    <OMP>
6480 WINDOW 0,5,79,24                                        <HK9>
6490 GOSUB 9930                                              <GUS>
6500 FOR I=1 TO PZ                                           <NM9>
6510 IF DI(I)=DK THEN NA$(I)="ZZZZ"                          <MSR>
6520 NEXT I                                                  <D14>
6530 IF DN>DZ THEN DZ=DN                                     <E8K>
6535 DB(2,S,DN)=1                                            <MI7>
6540 PRINT "{CLR,6DOWN}";TAB(12)"BITTE DISKETTE I
     NS LAUFWERK SCHIEBEN UND {RVSON}RETURN{RVOFF
     } DRUECKEN{SHF.SPACE}!"                                 <FL5>
6550 DO : GET KEY A$: LOOP UNTIL A$=CHR$(13)                 <BC1>
6570 PRINT "{CLR,6DOWN}";TAB(25)"DAS DIRECTORY WI
     RD GELESEN{DOWN}"                                       <HUB>
6580 OPEN 1,8,0,"$0": I1=0: I2=1                             <4KD>
6590 IF DS<>0 THEN PRINT "{CLR,6DOWN}";TAB(25)"{R
     VSON}DISK - STATUS:{RVOFF} ";DS$: GET KEY A$
     : DB(2,S,DN)=0: RETURN                                  <I0I>
6600 GET #1,A$,B$                                            <ETK>
6610 DO                                                      <5CN>
6620 GET #1,A$,B$                                            <ATK>
6630 GET #1,A$,B$                                            <8TK>
6640 C=0                                                     <PR3>
6650 IF A$<>"" THEN C=ASC(A$)                                <38O>
6660 IF B$<>"" THEN C=C+ASC(B$)*256                          <KKE>
6670 BL$=MID$(STR$(C),2)                                     <FA0>
6680 GET #1,B$: IF ST<>0 THEN 7020                           <1U2>
6690 IF B$<>CHR$(34) THEN 6680                               <D5A>
6700 DO                                                      <6CR>
6710 GET #1,B$                                               <TJK>
6720 IF B$=CHR$(34) THEN EXIT                                <ECK>
6730 N$=N$+B$                                                <FDM>
6740 LOOP                                                    <OB9>
6750 DO                                                      <3CF>
6752 GET #1,B$                                               <SRK>
6756 LOOP WHILE B$=CHR$(32)                                  <VFI>
6760 C$=""                                                   <K61>
6770 DO                                                      <1C7>
6772 C$=C$+B$: GET #1,B$                                     <KVE>
6774 LOOP WHILE B$<>""                                       <I6S>
6780 TY$=LEFT$(C$,3)                                         <69I>
6790 PRINT TAB(25)"GELESENE - FILES ";I1;"{UP}"              <C00>
6800 I1=I1+1                                                 <PFD>
6810 IF I1=1 THEN N$="": PZ=PZ+1: GOTO 6620                  <DM8>
6820 IF I2=1 THEN BEGIN                                      <LJA>
6830 NA$(PZ)=N$: DI(PZ)=DK: I2=0                             <7DC>
6840 FOR A=1 TO 6                                            <TI1>
6850 IF TY$=TL$(A) THEN TY(PZ)=A                             <OGV>
6860 NEXT A                                                  <SV0>
6870 IF TY(PZ)=0 THEN TY(PZ)=7                               <H9E>
6880 BEND                                                    <PRE>
6890 IF TR=0 THEN BL(PZ)=BL(PZ)+VAL(BL$)                     <OKP>
6900 IF TR=0 AND N$="----------------" THEN PZ=PZ
     +1: I2=1                                                <BSG>
6910 IF TR=1 THEN BEGIN                                      <FVB>
6920 BL(PZ)=VAL(BL$)                                         <LMC>
6930 I2=1                                                    <V7L>
6970 PZ=PZ+1                                                 <GKF>
6980 BEND                                                    <HQE>
6990 N$=""                                                   <MM9>
7000 IF ST<>0 THEN EXIT                                      <RH3>
7010 LOOP                                                    <4CP>
7020 CLOSE 1                                                 <DCN>
7030 DB(1,S,DN)=VAL(BL$)                                     <N7A>
7040 GOTO 7140                                               <18A>
7050 REM DATEN AENDERN                                       <4CH>
7060 IF SI=0 THEN SI=1                                       <E7O>
7070 WINDOW 0,0,79,24                                        <PT8>
7080 PRINT "{CLR}"                                           <CTM>
7090 PRINT TAB(29)"UCCCCCCCCCCCCCCCCCCCCCCI"                 <KFD>
7100 PRINT TAB(29)"B{5SPACE}DATEN AENDERN{6SPACE}
     B"                                                      <2QD>
7110 PRINT TAB(29)"JCCCCCCCCCCCCCCCCCCCCCCK"                 <PVF>
7120 WINDOW 0,5,79,24                                        <H4B>
7130 GOSUB 9930: PRINT "{CLR}"                               <9CP>
7140 I=0: LI=0: WINDOW 0,4,79,24                             <4QO>
7150 PRINT "{CLR,DOWN}"                                      <MQF>
7160 PRINT "DISKETTE NR. :";DN;TAB(30)"SEITE : ";
     S$;TAB(57)"FREIE BLOECKE :";DB(1,S,DN)                  <TG1>
7170 PRINT "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"              <3QB>
7180 PRINT "{DOWN,21SPACE}MOEGLICHE PROGRAMMARTEN
     {28SPACE}TYPEN"                                         <2S1>
7190 PRINT "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCCCCCCCCCCC{4SPACE}CCCCCCC
     CC "                                                    <E9U>
7200 PRINT " (01) ";AL$(1);TAB(27)"(07) ";AL$(7);
     TAB(49)"(13) ";AL$(13);TAB(71);"(1) ";TL$(1)            <B4N>
7210 PRINT " (02) ";AL$(2);TAB(27)"(08) ";AL$(8);
     TAB(49)"(14) ";AL$(14);TAB(71);"(2) ";TL$(2)            <PCA>
7220 PRINT " (03) ";AL$(3);TAB(27)"(09) ";AL$(9);
     TAB(49)"(15) ";AL$(15);TAB(71);"(3) ";TL$(3)            <NG2>
7230 PRINT " (04) ";AL$(4);TAB(27)"(10) ";AL$(10)
     ;TAB(49)"(16) ";AL$(16);TAB(71);"(4) ";TL$(4
     )                                                       <NQJ>
7240 PRINT " (05) ";AL$(5);TAB(27)"(11) ";AL$(11)
     ;TAB(49)"(17) ";AL$(17);TAB(71);"(5) ";TL$(5
     )                                                       <2K3>
7250 PRINT " (06) ";AL$(6);TAB(27)"(12) ";AL$(12)
     ;TAB(49)"(18) ";AL$(18);TAB(71);"(6) ";TL$(6
     );"{DOWN}"                                              <5UO>
7260 PRINT TAB(15)"{7SPACE}NAME{9SPACE}LAENGE{8SP
     ACE}ART{10SPACE}TYP"                                    <3U9>
7270 PRINT TAB(15)"UCCCCCCCCCCCCCCCCCCRCCCCCCRCCC
     CCCCCCCCCCCCCCRCCCCCI"                                  <49J>
7280 PRINT TAB(15)"B{18SPACE}B{6SPACE}B{18SPACE}B
     {5SPACE}B"                                              <1MR>
7290 PRINT TAB(15)"JCCCCCCCCCCCCCCCCCCFCCCCCCFCCC
     CCCCCCCCCCCCCCFCCCCCK"                                  <5HI>
7300 FOR IC=1 TO PZ                                          <B6F>
7310 IF DI(IC)=DK AND NA$(IC)<>"ZZZZ" THEN BEGIN             <TJ7>
7320 WINDOW 17,20,33,20: PRINT "{CLR}";NA$(IC);              <BRQ>
7330 WINDOW 36,20,40,20: PRINT "{CLR}";MID$(STR$(
     BL(IC)),2);                                             <MF4>
7340 IF AL$(AR(IC))="" THEN AR(IC)=19                        <9TV>
7350 WINDOW 43,20,59,20: PRINT "{CLR}";AL$(AR(IC)
     );                                                      <6NC>
7360 WINDOW 62,20,65,20: PRINT "{CLR}";TL$(TY(IC)
     );                                                      <82D>
7370 IF I=0 THEN BEGIN                                       <FO6>
7380 WINDOW 11,23,69,24                                      <BG7>
7390 PRINT "{CLR}WOLLEN SIE DIE DIE ANZAHL DER FR
     EIEN BLOECKE AENDERN (J/N)?";                           <ONG>
7400 GET KEY A$: I=1                                         <FO7>
7410 PRINT "{CLR}"                                           <CLM>
7420 IF A$="J" THEN BEGIN                                    <9H6>
7430 WINDOW 57,6,77,6: PRINT CHR$(15);"{RED}FREIE
      BLOECKE : {BLACK}";CHR$(143);                          <BRA>
7440 A=73: B=77: C=4: D=48: E=57: G=6: D$=STR$(DB
     (1,S,DN)): GOSUB 9520: DB(1,S,DN)=VAL(RIGHT$
     (D$,I1))                                                <M52>
7450 WINDOW 57,6,79,6: PRINT "FREIE BLOECKE :";DB
     (1,S,DN);                                               <9PU>
7460 BEND                                                    <HIE>
7470 BEND                                                    <LIU>
7480 WINDOW 25,22,60,24                                      <D0V>
7490 PRINT "{CLR,4SPACE}(A) - AENDERN"                       <QLS>
7500 PRINT "{4SPACE}(L) - LOESCHEN"                          <QLC>
7510 PRINT "(SPACE) - NAECHSTER FILE";                       <TIM>
7520 GET KEY A$                                              <3GD>
7530 IF A$<>"A" AND A$<>"L" AND A$<>" " THEN GOTO
      7520                                                   <0BN>
7540 PRINT "{CLR}"                                           <DDM>
7550 IF A$=" " THEN LI=1                                     <2D3>
7560 IF A$="L" THEN SI=2: NA$(IC)="ZZZZ": GOTO 80
     30                                                      <FND>
7570 IF A$="A" THEN BEGIN                                    <FH8>
7575 LI=1                                                    <DTV>
7580 REM NAMENSEINGABE                                       <5KM>
7590 WINDOW 15,19,34,22: PRINT CHR$(15);MR$(1);CH
     R$(143);                                                <10A>
7600 WINDOW 17,20,33,20: PRINT "{BLACK,CLR}";NA$(
     IC);                                                    <HDE>
7610 A=17: B=33: C=16: D=32: E=95: G=20: D$=NA$(I
     C): GOSUB 9520                                          <TJP>
7620 IF NA$(IC)<>D$ THEN SI=2                                <2K3>
7630 NA$(IC)=D$                                              <2K2>
7640 WINDOW 15,19,34,22: PRINT MS$(1);                       <4UK>
7650 WINDOW 17,20,33,20: PRINT "{BLACK,CLR}";NA$(
     IC);                                                    <5FU>
7660 REM LAENGENEINGABE                                      <7KU>
7670 WINDOW 34,19,41,22: PRINT CHR$(15);MR$(2);CH
     R$(143);                                                <99M>
7680 WINDOW 36,20,40,20: PRINT "{BLACK,CLR}";MID$
     (STR$(BL(IC)),2);                                       <F01>
7690 A=36: B=40: C=4: D=48: E=57: G=20: D$=STR$(B
     L(IC)): GOSUB 9520: BL(IC)=VAL(RIGHT$(D$,I1)
     )                                                       <T75>
7700 WINDOW 34,19,41,22: PRINT MS$(2);                       <CBM>
```

```
7710 WINDOW 36,20,40,20: PRINT "{BLACK,CLR}";RIGH
     T$(D$,I1);                                    <0J2>
7720 REM ARTEINGABE                                <640>
7730 WINDOW 41,19,60,22: PRINT CHR$(15);MR$(1);CH
     R$(143);                                      <BHE>
7740 WINDOW 43,20,59,20: PRINT "{BLACK,CLR}";AL$(
     AR(IC));                                      <FPB>
7750 DO                                            <7CV>
7760 D$=""                                         <Q3P>
7770 WINDOW 43,20,59,20: PRINT "{BLACK,CLR}";AL$(
     AR(IC));                                      <F1B>
7780 FOR I2=1 TO 2                                 <PMG>
7790 GET KEY A$: A=ASC(A$)                         <H38>
7800 IF A=20 THEN D$="": GOTO 7780                 <N5N>
7810 IF A=13 AND I2=1 THEN EXIT                    <VLJ>
7820 IF A>57 OR A<48 THEN GOTO 7790                <HP3>
7830 D$=D$+A$                                      <TJ5>
7840 NEXT I2                                       <TQG>
7850 IF VAL(D$)>18 OR VAL(D$)=0 THEN GOTO 7780     <G8D>
7860 AR(IC)=VAL(D$): WINDOW 43,20,59,20: PRINT "{
     CLR}";AL$(AR(IC));                            <TOA>
7870 LOOP                                          <S7P>
7880 WINDOW 41,19,60,22: PRINT MS$(3);             <L8J>
7890 WINDOW 43,20,59,20: PRINT "{BLACK,CLR}";AL$(
     AR(IC));                                      <B1B>
7900 REM TYPENEINGABE                              <1K6>
7910 WINDOW 60,19,66,22: PRINT CHR$(15);MR$(4);CH
     R$(143);                                      <9OA>
7920 WINDOW 62,20,65,20: PRINT "{BLACK,CLR}";TL$(
     TY(IC));                                      <GS9>
7930 DO                                            <TFN>
7940 GET KEY A$: A=ASC(A$)                         <01C>
7950 IF A=13 THEN EXIT                             <RA0>
7960 IF A>54 OR A<49 THEN GOTO 7940                <NHK>
7970 WINDOW 62,20,65,20: TY(IC)=ASC(A$)-48: PRINT
      "{CLR}";TL$(TY(IC));                         <JRR>
7980 LOOP                                          <819>
7990 WINDOW 60,19,66,22: PRINT MS$(4);             <E1U>
8000 WINDOW 62,20,65,20: PRINT "{BLACK,CLR}";TL$(
     TY(IC));                                      <RS9>
8010 BEND                                          <TBU>
8020 BEND                                          <HAE>
8030 NEXT IC                                       <198>
8040 WINDOW 20,23,60,24                            <OAF>
8050 PRINT "{CLR}KEINE WEITEREN EINTRAGUNGEN VORH
     ANDEN"                                        <F7H>
8055 GET KEY A$                                    <FCD>
8060 DB(2,S,DN)=LI                                 <MUG>
8070 RETURN                                        <QI7>
8080 REM DATEN EINFUEGEN                           <850>
8090 PRINT "{CLR}"                                 <3LM>
8100 PRINT TAB(27)"UCCCCCCCCCCCCCCCCCCCI"          <KDS>
8110 PRINT TAB(27)"B{3SPACE}DATEN EINFUEGEN{3SPAC
     E}B"                                          <611>
8120 PRINT TAB(27)"JCCCCCCCCCCCCCCCCCCCK"          <L1U>
8130 WINDOW 0,5,79,24: GOSUB 9930                  <CV1>
8140 PZ=PZ+1: NA$(PZ)="NEUER EINTRAG": TY(PZ)=7:
     AR(PZ)=19: DI(PZ)=DK: SI=2                    <D0N>
8150 IF DK>DZ THEN DZ=DK                           <9D0>
8155 DB(2,S,DN)=1                                  <EA4>
8160 GOTO 7140                                     <E8A>
8170 REM PROGRAMMARTEN AENDERN                     <BDD>
8175 DO                                            <NQT>
8180 IF SI=0 THEN SI=1                             <D7K>
8190 PRINT "{CLR}"                                 <35M>
8200 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCI"     <BLK>
8210 PRINT TAB(25)"B{2SPACE}PROGRAMMARTEN AENDERN
     {2SPACE}B"                                    <AV7>
8220 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCK"     <L5M>
8230 PRINT "{DOWN,22SPACE}AKTUELLE PROGRAMMARTEN"  <E36>
8240 PRINT "CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"            <LV1>
8250 FOR I=1 TO 6                                  <B61>
8260 PRINT " (";I;") ";TAB(8)AL$(I);TAB(24)" (";I
     +6;") ";TAB(32)AL$(I+6);TAB(48)" (";I+12;")
     ";TAB(56)AL$(I+12)                            <L1V>
8270 NEXT I                                        <KI4>
8280 PRINT "{DOWN}ART - NUMMER{3SPACE}: ";         <OVN>
8290 DO : L=2: A1=48: A2=57: GOSUB 10100: LOOP WH
     ILE VAL(E$)>18                                <2K7>
8300 IF VAL(E$)=0 THEN EXIT                        <O7H>
8310 I=VAL(E$)                                     <4NB>
8320 PRINT : PRINT "{DOWN}PROGRAMM - ART : ";      <SHV>
8330 L=16: A1=32: A2=95: GOSUB 10100               <BTM>
8340 AL$(I)=E$                                     <8Q0>
8345 LOOP                                          <II1>
8350 RETURN                                        <QQ7>
8360 REM PROGRAMMTYPEN AENDERN                     <A58>
8365 DO                                            <IQ9>
8370 IF SI=0 THEN SI=1                             <TJM>
8380 WINDOW 0,0,79,24: PRINT "{CLR}"               <I59>
8390 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCI"     <E5K>
8400 PRINT TAB(25)"B{2SPACE}PROGRAMMTYPEN AENDERN
     {2SPACE}B"                                    <UBF>
8410 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCK"     <ILM>
8420 WINDOW 25,5,79,24                             <OFL>
8430 PRINT "{DOWN,2SPACE}AKTUELLE PROGRAMMTYPEN"   <RMG>
8440 PRINT "CCCCCCCCCCCCCCCCCCCCCCCC"              <HPA>
8450 FOR I=1 TO 6                                  <D61>
8460 PRINT TAB(8)"(";I;") ";TL$(I)                 <OPH>
8470 NEXT I                                        <KU4>
8480 PRINT "{DOWN}TYP - NUMMER{3SPACE}: ";CHR$(15
     );"B{LEFT}";CHR$(143);                        <OK9>
8490 DO : GET KEY E$: LOOP WHILE VAL(E$)>6         <F3B>
8500 PRINT E$: IF VAL(E$)=0 THEN EXIT              <KGS>
8510 I=VAL(E$)                                     <2NB>
8520 PRINT "{DOWN}PROGRAMM - TYP : ";              <9F8>
8530 L=3: A1=32: A2=95: GOSUB 10100                <QNH>
8540 TL$(I)=E$                                     <CE0>
8545 LOOP                                          <2K1>
8550 RETURN                                        <Q27>
8560 REM DATEN SORTIEREN UND SPEICHERN             <954>
8570 IF PZ=0 OR SI=0 THEN RETURN                   <118>
8580 IF SI=1 THEN GOTO 9240                        <RR8>
8590 SZ=INT(PZ/2)+PZ-1                             <KR0>
8600 PRINT "{CLR,7DOWN}"                           <L60>
8610 PRINT TAB(18)"UCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCI"                            <ET0>
8620 PRINT TAB(18)"B{47SPACE}B"                    <HD1>
8630 PRINT TAB(18)"B{11SPACE}DIE DATEN WERDEN SOR
     TIERT{11SPACE}B"                              <JVF>
8640 PRINT TAB(18)"B{47SPACE}B"                    <HL2>
8650 PRINT TAB(18)"B{5SPACE}ES SIND";SZ;"RECHENSC
     HRITTE NOTWENDIG";TAB(66);"B"                 <9EJ>
8660 PRINT TAB(18)"B{47SPACE}B"                    <HT3>
8670 PRINT TAB(18)"B{47SPACE}B"                    <HP3>
8680 PRINT TAB(18)"B{47SPACE}B"                    <G54>
8690 PRINT TAB(18)"JCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCK"                            <8LP>
8700 PRINT "{4UP}"                                 <CAB>
8710 REM HEAP SORT                                 <9T7>
8720 L=INT(PZ/2)+1: R=PZ                           <FVJ>
8730 DO WHILE L>1                                  <T0O>
8740 L=L-1: SZ=SZ-1                                <2RJ>
8750 PRINT TAB(30)"VERBLEIBENDE SCHRITTE : ";STR$
     (SZ);" {UP}"                                  <KQM>
8760 REM SIFT                                      <850>
8770 I=L: J=2*I: X1$=NA$(I): X2=TY(I): X3=BL(I):
     X4=DI(I): X5=AR(I)                            <1JN>
8780 PL=0                                          <5GT>
8790 DO                                            <NAV>
8800 IF J>R OR PL=1 THEN EXIT                      <5QK>
8810 IF J<R THEN IF NA$(J)<NA$(J+1) THEN J=J+1     <KPC>
8820 IF X1$>NA$(J) THEN PL=1: ELSE BEGIN           <E68>
8830 NA$(I)=NA$(J)                                 <MC2>
8840 TY(I)=TY(J)                                   <RFV>
8850 BL(I)=BL(J)                                   <E0R>
8860 DI(I)=DI(J)                                   <39V>
8870 AR(I)=AR(J)                                   <K3R>
8880 I=J: J=2*I                                    <B3F>
8890 BEND                                          <T7U>
8900 LOOP                                          <OR9>
8910 NA$(I)=X1$: TY(I)=X2: BL(I)=X3: DI(I)=X4: AR
     (I)=X5                                        <DM3>
8920 REM SIFT END                                  <E5O>
8930 LOOP                                          <KQP>
8940 DO WHILE R>1                                  <914>
8950 Z$=NA$(R): NA$(R)=NA$(L): NA$(L)=Z$           <CDN>
8960 Z=TY(R): TY(R)=TY(L): TY(L)=Z                 <UKJ>
8970 Z=BL(R): BL(R)=BL(L): BL(L)=Z                 <SNQ>
8980 Z=DI(R): DI(R)=DI(L): DI(L)=Z                 <T93>
8990 Z=AR(R): AR(R)=AR(L): AR(L)=Z                 <VB2>
9000 PRINT TAB(30)"VERBLEIBENDE SCHRITTE : ";STR$
     (SZ);" {UP}"                                  <8T6>
9010 R=R-1: SZ=SZ-1                                <9JI>
9020 REM SIFT                                      <BLE>
9030 I=L: J=2*I: X1$=NA$(I): X2=TY(I): X3=BL(I):
     X4=DI(I): X5=AR(I)                            <23N>
9040 PL=0                                          <TJT>
9050 DO                                            <HA7>
9060 IF J>R OR PL=1 THEN EXIT                      <1U4>
9070 IF J<R THEN IF NA$(J)<NA$(J+1) THEN J=J+1     <GTS>
9080 IF X1$>NA$(J) THEN PL=1: ELSE BEGIN           <FQF>
9090 NA$(I)=NA$(J)                                 <6G1>
9100 TY(I)=TY(J)                                   <VBV>
9110 BL(I)=BL(J)                                   <A4R>
9120 DI(I)=DI(J)                                   <7DV>
9130 AR(I)=AR(J)                                   <G7R>
9140 I=J: J=2*I                                    <63F>
9150 BEND                                          <LAU>
9160 LOOP                                          <GI9>
9170 NA$(I)=X1$: TY(I)=X2: BL(I)=X3: DI(I)=X4: AR
     (I)=X5                                        <CI7>
9180 REM SIFT END                                  <9L6>
9190 LOOP                                          <CLP>
9200 REM HEAP SORT END                             <A58>
9210 DO WHILE NA$(PZ)="ZZZZ"                       <2R4>
9220 PZ=PZ-1                                       <KO9>
9230 LOOP                                          <SJP>
9240 PRINT "{CLR,9DOWN}";TAB(28)"DIE DATEN WERDEN
      GESPEICHERT"                                 <6N5>
9280 PRINT TAB(28)"{DOWN}DATEN DISKETTE EINLEGEN"  <2P6>
9285 GET KEY A$                                    <C4D>
9290 PRINT TAB(28)"{UP,23SPACE}"                   <5C3>
9300 SCRATCH (DN$)                                 <5GN>
9310 DOPEN #1,(DN$),W                              <P1F>
9311 IF DS<>0 THEN DCLOSE #1: PRINT TAB(28)"{DOWN
     ,RVSON}DISKETTENSTATUS{RVOFF}:";DS$: GET KEY
      A$: GOTO 9240                                <ERQ>
9320 FOR A=1 TO 6                                  <BI1>
9330 PRINT#1,TL$(A)                                <UIP>
9340 NEXT A                                        <TN0>
9350 FOR A=1 TO 18                                 <MOP>
```

Listing 2. (Fortsetzung)

```
9360 PRINT#1,AL$(A)                                   <QG8>
9370 NEXT A                                           <5K0>
9380 PRINT#1,DZ                                       <GDF>
9390 FOR A=1 TO DZ                                    <BQT>
9400 PRINT#1,DB(1,1,A)                                <L62>
9402 PRINT#1,DB(1,2,A)                                <MMC>
9404 PRINT#1,DB(2,1,A)                                <O7M>
9406 PRINT#1,DB(2,2,A)                                <RND>
9410 NEXT A                                           <5S0>
9420 PRINT#1,PZ                                       <HDR>
9430 FOR A=1 TO PZ                                    <AQ9>
9440 PRINT#1,NA$(A)                                   <SCJ>
9450 PRINT#1,TY(A)                                    <PRM>
9460 PRINT#1,BL(A)                                    <OSI>
9470 PRINT#1,DI(A)                                    <PTM>
9480 PRINT#1,AR(A)                                    <GJJ>
9490 NEXT A                                           <5K0>
9500 DCLOSE #1                                        <795>
9510 RETURN                                           <Q27>
9520 REM EINGABEMODUL                                 <954>
9530 I1=0                                             <E1J>
9540 DO                                               <QBB>
9550 GET KEY A$                                       <C8D>
9560 IF A$=CHR$(20) AND LEN(D$)=0 THEN LOOP           <UNS>
9570 IF A$=CHR$(20) THEN PRINT "{LEFT} {LEFT}";:
     D$=LEFT$(D$,LEN(D$)-1): LOOP                     <T9Q>
9580 IF A$=CHR$(13) AND I1=0 THEN I1=LEN(D$): RET
     URN                                              <96H>
9590 IF A$=CHR$(13) THEN RETURN                       <SEH>
9600 IF ASC(A$)<D OR ASC(A$)>E THEN LOOP              <E0M>
9610 IF I1=0 THEN WINDOW A,G,B,G: D$="": PRINT "{
     CLR}";                                           <QV6>
9630 IF LEN(D$)=C THEN LOOP                           <UOR>
9640 PRINT A$;                                        <TIB>
9650 D$=D$+A$: I1=I1+1                                <VCS>
9660 LOOP                                             <8L9>
9730 REM DISKETTENKOMMANDO                            <BD1>
9740 PRINT "{CLR}"                                    <2DM>
9750 PRINT TAB(25)"UCCCCCCCCCCCCCCCCCCCCCCCCCI"       <E5K>
9760 PRINT TAB(25)"B{4SPACE}DISKETTENKOMMANDO{4SP
     ACE}B"                                           <REJ>
9770 PRINT TAB(25)"JCCCCCCCCCCCCCCCCCCCCCCCCCK"       <GLM>
9780 PRINT "{7DOWN}KOMMANDO <";                       <SLB>
9790 L=40: A1=32: A2=95: GOSUB 10100                  <75M>
9800 OPEN 1,8,15,E$: CLOSE 1                          <OB2>
9810 PRINT : PRINT "{2DOWN,RVSON}DISK-STATUS{RVOF
     F}:";DS$                                         <R78>
9820 GET KEY A$                                       <B0D>
9830 RETURN                                           <RI7>
9840 REM DIRECTORY                                    <E50>
9850 PRINT "{CLR}"                                    <5LM>
9860 PRINT TAB(28)"UCCCCCCCCCCCCCCCCCCCI"             <JDS>
9870 PRINT TAB(28)"B{6SPACE}DIRECTORY{6SPACE}B"       <748>
9880 PRINT TAB(28)"JCCCCCCCCCCCCCCCCCCCK"             <IDU>
9890 WINDOW 26,6,54,24                                <CP6>
9900 D=1: DIRECTORY : D=0                             <14Q>
9910 PRINT "{DOWN} TASTE DRUECKEN": GET KEY A$        <DHO>
9920 RETURN                                           <BK7>
9930 REM DISKETTEN ANGABE                             <FDT>
9940 PRINT "{CLR}BITTE GEBEN SIE NUMMER UND SEITE
      DER DISKETTE AN !"                              <NEP>
9950 FOR I=1 TO 80: PRINT "C";: NEXT I                <LFI>
9960 PRINT : PRINT "{DOWN}DISKETTE (1-999) :";        <OFG>
9970 L=3: A1=48: A2=57: GOSUB 10100                   <PNL>
9980 IF VAL(E$)=0 THEN GOTO 9940                      <KUB>
9990 PRINT : PRINT "{DOWN}SEITE (A/B){6SPACE}:";C
     HR$(15);"B{LEFT}";CHR$(143);                     <7TE>
10000 DO                                              <PV3>
10010 GET KEY S$                                      <04U>
10020 IF S$="A" OR S$="B" THEN EXIT                   <VSB>
10025 LOOP                                            <K61>
10030 PRINT S$                                        <D5P>
10040 PRINT "{3DOWN}EINGABE IN ORDNUNG ?"             <IMB>
10050 GET KEY A$                                      <5KC>
10060 IF A$="N" THEN GOTO 9940                        <QHM>
10070 IF S$="B" THEN S=0: ELSE S=1                    <HAR>
10080 DK=2*VAL(E$)-S: DN=VAL(E$): IF S=0 THEN S=2     <4OV>
10090 RETURN                                          <0U7>
10100 REM EINGABE II                                  <0O2>
10110 E$="": PRINT CHR$(15);"B{LEFT}";CHR$(143);      <OO9>
10120 DO                                              <PV3>
10130 GET KEY A$                                      <4KC>
10140 A=ASC(A$)                                       <T2O>
10150 IF A=13 AND E$="" THEN LOOP                     <B20>
10160 IF A=13 THEN PRINT " ";: RETURN                 <D2G>
10170 IF A=20 AND LEN(E$)=0 THEN LOOP                 <SEF>
10180 IF A=20 THEN E$=LEFT$(E$,LEN(E$)-1): PRINT
      "{LEFT}";CHR$(15);"B {2LEFT}";CHR$(143);: L
      OOP                                             <2P5>
10190 IF LEN(E$)=L THEN LOOP                          <EER>
10200 IF A<A1 OR A>A2 THEN LOOP                       <PAF>
10210 PRINT A$;CHR$(15);"B{LEFT}";CHR$(143);          <H5H>
10220 E$=E$+A$                                        <FLC>
10230 LOOP                                            <A5P>
10240 REM PROGRAMM NAMEN AUSGEBEN                     <080>
10250 IF NA$(I)=NA$(I+1) THEN BEGIN                   <LVO>
10260 BL=BL(I): DL=DI(I): DH=DI(I): NA$=NA$(I)        <10S>
10270 DO                                              <QVF>
10280 I=I+1                                           <FGT>
10290 BL=BL+BL(I)                                     <COL>
10300 IF DI(I)<DL THEN DL=DI(I)                       <OFT>
10310 IF DI(I)>DH THEN DH=DI(I)                       <G9T>
10320 LOOP WHILE NA$(I)=NA$(I+1)                      <RAO>
10330 IF INT(DL/2)<>DL/2 THEN SL$="A": DL=(DL+1)/
      2: ELSE SL$="B": DL=DL/2                        <R1G>
10340 IF INT(DH/2)<>DH/2 THEN SH$="A": DH=(DH+1)/
      2: ELSE SH$="B": DH=DH/2                        <S0O>
10350 DL$=STR$(DL): DH$=STR$(DH)                      <SP0>
10360 DO WHILE LEN(DL$)<3 : DL$=" "+DL$: LOOP         <E1B>
10370 DO WHILE LEN(DH$)<3 : DH$=" "+DH$: LOOP         <V3C>
10380 D$=DL$+" "+SL$+" -"+DH$+" "+SH$                 <SUV>
10390 BEND : ELSE BEGIN                               <RGG>
10400 IF INT(DI(I)/2)<>DI(I)/2 THEN S$="A": D=(DI
      (I)+1)/2: ELSE S$="B": D=DI(I)/2                <DC7>
10410 D$=STR$(D)                                      <574>
10420 DO WHILE LEN(D$)<3 : D$=" "+D$: LOOP            <63I>
10430 D$=D$+" "+S$: BL=BL(I): NA$=NA$(I)              <ROT>
10440 BEND                                            <FPE>
10450 IF DM$<>"" THEN RETURN                          <LKC>
10460 IF IC=0 THEN WINDOW 0,8,79,21: PRINT "{CLR}
      ";: IC=1                                        <MB2>
10470 PRINT NA$;TAB(20)BL;TAB(30)AL$(AR(I));TAB(5
      1)"(";D$;")";TAB(72)TL$(TY(I))                  <2CD>
10480 GET A$: IF A$<>"" THEN GET KEY A$               <VQ8>
10490 RETURN                                          <0E7>
10500 REM FEHLERBEHANDLUNG                            <20A>
10510 TRAP 10500                                      <358>
10520 IF D=1 THEN RESUME NEXT                         <Q71>
10530 IF ER=5 AND DS=0 THEN PRINT : PRINT TAB(30)
      "{DOWN}DRUCKER EINSCHALTEN !": GET KEY A$:
      CLOSE 1: CLOSE 2: CLOSE 3: RETURN               <2U8>
10540 RESUME                                          <H41>
```

Listing 2. »Disk-Manager 128«, die komfortable Diskettenverwaltung. Bitte mit dem Checksummer (Seite 158) eingeben.

```
10 FAST                                               <S3J>
20 GRAPHIC 5,1                                        <0EQ>
30 COLOR 6,16                                         <OBA>
40 DZ=100: PZ=300                                     <ETC>
50 DIM NA$(PZ): DIM TY(PZ): DIM AL$(18)               <LS6>
60 DIM BL(PZ): DIM DI(PZ) : DIM AR(PZ): DIM FB(2,
   2,DZ)                                              <J8J>
70 PRINT "{DOWN,BLACK,24SPACE}KOPIERPROGRAMM FUER
    SYSTEMDATEIEN"                                    <E6U>
80 PRINT "{23SPACE}EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
   EEE"                                               <B26>
90 INPUT "{DOWN}QUELL - NAME : ";SN$                  <NLN>
100 INPUT "{DOWN}ZIEL - NAME{2SPACE}: ";DN$           <E0N>
110 DOPEN #1,(SN$)                                    <E4A>
120 FOR I=1 TO 6                                      <RQ0>
130 INPUT#1,TL$(I)                                    <QB8>
140 NEXT I                                            <RN4>
150 FOR I=1 TO 18                                     <MCQ>
160 INPUT#1,AL$(I)                                    <U9P>
170 NEXT I                                            <3K4>
180 INPUT#1,DZ                                        <S0C>
190 FOR I=1 TO DZ                                     <BEU>
200 INPUT#1,FB(1,1,I)                                 <58J>
210 INPUT#1,FB(1,2,I)                                 <6OT>
220 INPUT#1,FB(2,1,I)                                 <897>
230 INPUT#1,FB(2,2,I)                                 <BP9>
240 NEXT I                                            <BH4>
250 INPUT#1,PZ                                        <I0O>
260 FOR I=1 TO PZ                                     <7EA>
270 INPUT#1,NA$(I)                                    <810>
280 INPUT#1,TY(I)                                     <GJ2>
290 INPUT#1,BL(I)                                     <1K6>
300 INPUT#1,DI(I)                                     <GT3>
310 INPUT#1,AR(I)                                     <9V7>
320 NEXT I                                            <BL4>
330 DCLOSE #1                                         <TL5>
340 PRINT "{2DOWN}INSERT DESTINATION - DISK": GET
     KEY A$                                           <H6K>
350 SCRATCH (DN$)                                     <20P>
360 DOPEN #1,(DN$),W                                  <5EU>
370 FOR I=1 TO 6                                      <RA0>
380 PRINT#1,TL$(I)                                    <2FA>
390 NEXT I                                            <JU4>
400 FOR I=1 TO 18                                     <PCQ>
410 PRINT#1,AL$(I)                                    <69R>
420 NEXT I                                            <RR4>
430 PRINT#1,DZ                                        <HHC>
440 FOR I=1 TO DZ                                     <CEU>
450 PRINT#1,FB(1,1,I)                                 <7G8>
460 PRINT#1,FB(1,2,I)                                 <502>
470 PRINT#1,FB(2,1,I)                                 <AHS>
480 PRINT#1,FB(2,2,I)                                 <E1E>
490 NEXT I                                            <3G4>
500 PRINT#1,PZ                                        <LHO>
510 FOR I=1 TO PZ                                     <EEA>
520 PRINT#1,NA$(I)                                    <GD3>
530 PRINT#1,TY(I)                                     <SR2>
540 PRINT#1,BL(I)                                     <TO6>
550 PRINT#1,DI(I)                                     <SP2>
560 PRINT#1,AR(I)                                     <LR6>
570 NEXT I                                            <3S4>
580 DCLOSE #1                                         <SD5>
```

Listing 3. Das Programm zu Erstellung von Sicherheitskopien der mit »Disk-Manager 128« erstellten Dateien.

Bringen Sie Ordnung in Ihr Video-Archiv



Wer Video-Filme sammelt, braucht früher oder später ein Verzeichnis, das alle Titel mit den wichtigsten Daten enthält und ein rasches Aufsuchen einzelner Filme ermöglicht. Das hier vorgestellte »Videoarchiv-System« kann noch viel mehr. Es verwaltet bis zu 1100 Titel und macht dank komfortabler Sortier- und Druckroutinen die Dokumentation Ihrer Video-Sammlung zum Vergnügen.

Natürlich kann man seine Video-Kassetten auch mit einem universellen Datei-Programm, wie Datamat, verwalten. Wer sich aber die Mühe von Maskenaufbau, Definition von Kategorien, Erstellung von Drucklisten und andere Kleinigkeiten sparen will und stattdessen eine Komplettlösung sucht, die von Anfang an alle Wünsche berücksichtigt, der ist mit dem »Videoarchiv-System« bestens bedient. Das Programm arbeitet im 80-Zeichen-Modus des C128 und ist nach dem Prinzip einer index-sequentiellen Dateiverwaltung aufgebaut. Pro Diskette werden maximal 1100 Videofilme verwaltet, was auch für größere Sammlungen ausreichend sein dürfte. Für jeden Film stehen folgende Datenfelder (vergleiche Tabelle 1) zur Verfügung:

- Videotitel (Indexfeld, maximal 55 Zeichen)
- Kategorie (Indexfeld, 20 mögliche Kategorien)
- Kassettennummer (Indexfeld, maximal vier Stellen)
- Laufzeit in Minuten (drei Stellen)
- Farbfilm (ja/nein)
- Dolby-Rauschunterdrückung (ja/nein)
- Longplay (ja/nein)
- Stereo (ja/nein)
- Hifi-Ton (ja/nein)
- Bemerkungen (maximal 42 Zeichen)

In Zeichen	wird gespeichert	In Zeichen	wird gespeichert
1 bis 55	Titel	65	Dolby
56	Kategorie (a-t)	66	Longplay
57 bis 60	Kassettennummer	67	Stereo
61 bis 63	Laufzeit	68	Hifi
64	Farbe	69 bis 110	Bemerkungen

Tabelle 1. Der Aufbau eines Datensatzes im Videoarchiv-System. Ein Datensatz ist 110 Zeichen lang.

In Zeichen	wird gespeichert	In Zeichen	wird gespeichert
1 bis 15	Titel (Kurzfassung)	17 bis 20	Kassettennummer
16	Kategorie (a-t)	21 bis 24	Datensatznummer

Tabelle 2. Der Aufbau eines Indexstring beim Videoarchiv-System. Er hat eine Länge von 24 Zeichen.

Die drei genannten Indexfelder (siehe auch Tabelle 2) ermöglichen ein besonders rasches Auffinden einzelner Datensätze (maximal 2,5 Sekunden), da die Indexfelder ständig im Speicher des Computers stehen, was ja gerade den großen Vorteil der index-sequentiellen Dateiverwaltung ausmacht. Generell ist jedoch die Suche nach allen Feldinhalten sowie nach beliebigen Kombinationen möglich.

Schnelle Suche nach Filmen

Durch verschiedene Sicherheitsabfragen wird ein versehentlicher Datenverlust verhindert; viele Bedienungsfehler werden erkannt und abgefangen. Zur Vermeidung unnötiger Wartezeiten sortiert das Programm nur auf Anforderung sowie bei Beenden der Arbeit; dabei wird ein »Quicksort«-Algorithmus verwendet.

Das Programm verfügt für die Darstellung auf dem Bildschirm über einen eigenen, neuen Zeichensatz mit dem Schrifttyp »Orator«, der auf die DIN-Tastatur des C128 abgestimmt wurde. Für den Ausdruck der Ergebnisse auf Papier dürfen entweder Epson-kompatible Drucker oder die Commodore-Drucker MPS 801 und MPS 803 (letztere auch mit deutschen Umlauten!) verwendet werden. Für Geräte mit Centronics-Anschluß wurde eine softwaremäßige Schnittstelle integriert; es wird lediglich ein entsprechendes Userport-Kabel benötigt. Unterstützt werden aber auch folgende Interfaces: Görlitz VC-DIN, Görlitz VCEI, Data-Becker-Interface. Für die Ausgabe von Listen stehen vier Zeilen mit jeweils 40 Zeichen Text als »Benutzerdaten« zur Verfügung, die gespeichert werden und beim Ausdruck automatisch am Listenanfang erscheinen.

Integrierte Centronics-Schnittstelle

Der Betrieb ist sowohl mit einem als auch mit zwei Laufwerken vom Typ 1541/1570/1571 möglich, wobei die jeweils verwendete Konfiguration vom Programm automatisch festgestellt und korrekt angesprochen wird. Die Bedienung des Programms ist sehr einfach, weitgehend sogar selbsterklärend und erfolgt vollständig über Menüs. Zur Menüsteuerung dienen die *oberen* Cursortasten des C128 oder aber Joystick bzw. Maus am Control-Port 2. Dies ist vielleicht am Anfang etwas gewöhnungsbedürftig; wer damit anfänglich Probleme hat, kann mit Hilfe der HELP-Taste einen Hilfsbildschirm (Bild 1) abrufen.

Kurzbeschreibung der einzelnen Programmteile

»VIDEOARCHIV« (Listing 1): Dies ist die Basic-Version des Hauptprogrammes (Variablen-Übersicht in Tabelle 3). Die Programmservice-Diskette enthält außerdem die compilierte Version, die nicht nur kürzer, sondern auch deutlich schneller als der Basic-Quellcode ist. Wenn Sie das Programm selbst compilieren wollen, müssen Sie den Austro-Comp 128 verwenden, da das Programm wegen seiner komplizierten Variablenfelder von Basic 128 nicht akzeptiert wird.

»VIDEO-CHARACTERS« (Listing 2): Der veränderte Bildschirmzeichensatz »Orator« als MSE-Listing. Er wird vom Hauptprogramm nachgeladen.

»KATEGORIEN« (Listing 3): Dieses Teilprogramm erstellt die sequentielle Datei »video-katgorien«. Sie enthält die 20 vorgegebenen Standardkategorien, die aber nach individuellen Wünschen abgewandelt werden können.

Für jeden Drucker ein eigenes Programm

»VIDEO-CENTRONICS« (Listing 4): Die Software-Centronics-Schnittstelle, ebenfalls als MSE-Listing. Wenn Ihr Drucker nicht über ein Userportkabel angesteuert wird, benötigen Sie dieses Programm nicht. Die Zeilen 190 bis 220 und 2300 des Hauptprogramms dürfen dann nicht eingetippt werden.

»DRUCKERTAB.MPS« (Listing 5): Dieses kleine Programm erstellt die sequentiellen Dateien »video-mps 801« und »video-mps 803«, die vom Hauptprogramm eingelesen und in der Datei »video-standards« auf der Datendiskette gespeichert werden. Dabei handelt es sich um die Zeichentabellen für den intergrierten Codewandler. Listing 5 müssen Sie nur eingeben, wenn Sie einen der Drucker MPS-801 oder MPS-803 besitzen.

»DRUCKERTAB.EPSON« (Listing 6): Dieses Teilprogramm erfüllt den gleichen Zweck wie Listing 5, nur daß eine Druckertabelle für Epson-Drucker erstellt wird. Auch Listing 6 brauchen Sie nur abzutippen, wenn Sie einen

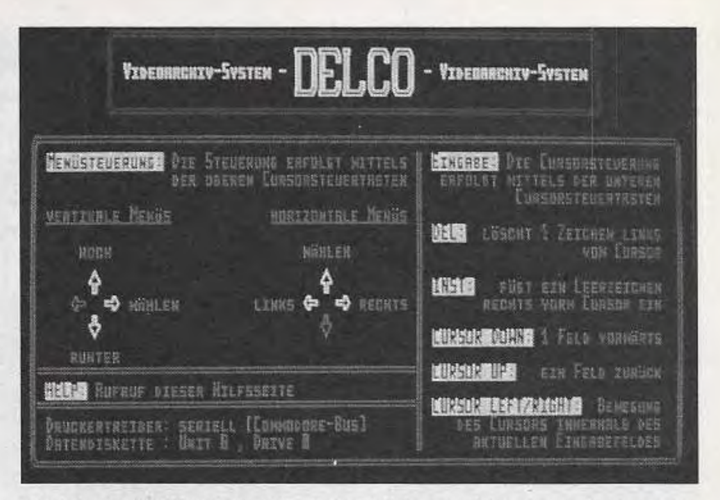


Bild 1. Der Hilfsbildschirm des Videoarchivs kann jederzeit aufgerufen werden.

Variable	Verwendung
te$(...)	Menütexte
te	Anzahl der Menütexte
by	Beginn des Menüs (Spalte)
f1%	oszillierendes Flag (bewirkt einen permanenten Wechsel zwischen den beiden Bildschirmen)
n%	Anzahl der Datensätze n%=0 : Datei ist leer
m%	Anzahl der Datensätze im Puffer
pu$(...)	Datenpuffer (Größe: 114)
pr$(...)	Druckertabelle (Größe: 255)
g$(...)	Indexstring (Größe: 1100)
d$	Aktueller Datensatz im Speicher
d	Nummer des aktuellen Datensatzes
z$	erlaubte Zeichen (Input-Routine)
sa%	Sekundäradresse des Druckers
cw%	Steuerflag für die Codewandlung cw%=1 : Codewandlung eingeschaltet cw%=0 : Codewandlung ausgeschaltet
p%	Anzahl der gedruckten vollständigen Datensätze (0 - 3)
dd%	Drive Datendiskette (0 oder 1)
ud%	Unit Datendiskette (8 oder 9)
dm%	Drive Programmdiskette (0 oder 1)
um%	Unit Programmdiskette (8 oder 9)
si	Steuerflag der Bildschirmwechselroutine si=0 : Bildschirm nicht sichtbar si=1 : Bildschirm sichtbar
bi	Steuerflag der Bildschirmwechselroutine bi=0 : Bildschirm »0« eingeschaltet bi=1 : Bildschirm »1« eingeschaltet
f1%	Flag, ob Datei »video-standards« geladen werden soll f1%=0 : nicht laden f1%=1 : laden
f2%	Flag, ob Indexdatei geladen wurde f2%=0 : noch nicht geladen f2%=1 : bereits geladen
f3%	Steuerflag für die »Input«-Routine
f4%	momentan bearbeitete Eingabefeldnummer
f5%	Flag für die Datenausgabe f5%=0 : Ausgabe auf Drucker f5%=1 : Ausgabe auf Bildschirm
f6%	Steuerflag für die Eingabemaske f6%=0 : normale Eingabe f6%=1 : Eingabe für »suchen, ändern, löschen« f6%=2 : Änderungsmodus f6%=3 : Kriteriensuche
f7%	Steuerflag für Suchmodus f7%=0 : Suchmodus eingeschaltet f7%=1 : Suchmodus ausgeschaltet
f8%	Offset-Flag für Druckerausgabe f8%=0 : kein Offset f8%=6 : Offset = 6

Tabelle 3. Die wichtigsten Variablen von »VIDEOARCHIV«



Bild 2. Nach dem Start erscheint das Hauptmenü des Videoarchiv-Systems



Bild 3. Vom Hauptmenü aus werden alle Funktionen des Programms angewählt



Bild 4. Das Druckmenü zum Drucken einer Titelliste zeigt die vielfältigen Möglichkeiten des Programms



Bild 5. Vom Dateipflege-Menü aus werden die leistungsfähigen Indexfunktionen angewählt

Epson- oder kompatiblen Drucker haben, egal ob er über ein Interface oder über ein Kabel am Userport angeschlossen ist.

Hinweise zur Bedienung

Geladen und gestartet wird das Programm mit RUN "VIDEO*"; es meldet sich nach kurzer Zeit mit dem Titelbild. Durch einen beliebigen Tastendruck gelangt man in das Startmenü. Bei diesem wie bei allen anderen vertikalen, also von oben nach unten angeordneten Menüs erfolgt die Bedienung mit Hilfe der *oberen* Cursortasten des C 128: < ↑ > rollt den revers dargestellten Balken schrittweise nach oben, < ↓ > bewegt ihn nach unten. Mit < → > wird die gewünschte Funktion gewählt (siehe auch Hinweise auf der durch die HELP-Taste aufrufbaren Hilfsseite). Eine analoge Steuerung ist auch mittels Joystick oder Maus am Control-Port 2 möglich, wobei zur Anwahl der Menüpunkte der Feuerknopf gedrückt werden muß.

Das Startmenü(Bild 2)

Wenn Sie noch keine Datendiskette angelegt haben, wählen Sie hier zunächst den Menüpunkt »Anlegen einer neuen Datei«. Das daraufhin erscheinende Menü hat nur zwei Punkte: »Standardkategorien« und »eigene Kategorien«. Sie können hier wählen, ob die Video-Kassetten nach den vom Programm vorgegebenen Kategorien verwaltet werden sollen oder ob Sie lieber Ihre eigenen Kategorien eingeben wollen.

Die Datendiskette wird vorbereitet

Wird »eigene Kategorien« gewählt, so fragt der Computer nach jeder Kategorie. Das nächste Menü dient dann noch der Auswahl des Druckers und erklärt sich selbst. Sollten Sie nur ein Laufwerk angeschlossen haben, so werden Sie jeweils zum Diskettenwechsel aufgefordert. Ist ein Doppellaufwerk angeschlossen, so erwartet das Programm die Programmdiskette in Laufwerk 0 und die Datendiskette in Laufwerk 1. Verwenden Sie zwei Einzellaufwerke, so muß sich das Programm in Laufwerk 8 und die Datendiskette in Laufwerk 9 befinden. Das Anlegen der Datendiskette dauert einige Zeit (ca. sieben Minuten), danach gelangen Sie wieder in das Startmenü. Dieses besteht aus vier weiteren Menüpunkten:

»Arbeit mit bestehender Datei«: Hiermit gelangen Sie in das Hauptmenü (siehe unten).

»Datendiskette wechseln«: Sollten Sie mehrere Datendisketten besitzen, so müssen Sie einen Diskettenwechsel über diesen Menüpunkt vornehmen.

»Festgelegte Daten ändern«: Dieser Punkt dient zur Änderung der Datei »video-standards« auf der Datendiskette. Hiermit ist also eine Modifizierung der verwendeten Film-Kategorien und der Benutzerdaten (=Titelleiste für den Ausdruck) sowie eine Neuwahl des Druckertreibers möglich.

»Video-Archiv verlassen«: Ein Abschluß der Arbeit mit dem Videoarchiv-System sollte immer über diesen Punkt erfolgen. Dabei wird der Index sortiert und gespeichert, anschließend erfolgt ein Reset. Achtung: Wird das Videoarchiv nicht auf diese Weise verlassen, erfolgt keine Speicherung des Index auf Diskette!. Die Datei geht dabei zwar nicht verloren, kann aber erst wieder verwendet werden, wenn der Index vom Programm rekonstruiert wurde, was sehr lange dauern kann.

Das Hauptmenü (Bild 3)

Bevor man erstmalig in das Hauptmenü gelangt, muß zunächst das aktuelle Datum in der Form »TTMMJJJJ« eingegeben werden. Danach besteht die Wahl zwischen sechs verschiedenen Menüpunkten.

DELCO VIDEO-ARCHIV | EINGABE: | DATUM: 14.03.1987 DATENSATZ: 0058

TITEL: AUF DEM HIGHWAY IST DIE HÖLLE LOS

KATEGORIE: A [ACTIONFILM]

CASS.NR.: 100 LAUFZEIT: 90 MINUTEN

FARBE: JA DOLBY: NEIN

LONGPLAY: NEIN STEREO: NEIN HIFI: JA

BEMERKUNGEN: MIT BURT REYNOLDS UND ROGER MOORE

HAUPTMENÜ SPEICHERN SPEICHERN & DRUCKEN KATEGORIEN STATISTIK

Bild 6. Hier wurde bereits ein Film mit allen benötigten Daten in die Eingabemaske eingetragen

»Titel eingeben«: Die Eingabemaske erscheint, und Sie können Ihre Eingaben zu einzelnen Filmtiteln durchführen. (Näheres siehe weiter unten.)

»Titel suchen, ändern, löschen«: Bei diesem Menüpunkt ist eine nachträgliche Bearbeitung bereits gespeicherter Daten möglich.

Komfortable Steuerung

»Titelliste auf Monitor«: Hier können Sie sich die in der Datei enthaltenen Titel auf dem Bildschirm anzeigen lassen. Wenn eine Bildschirmseite beschrieben ist, stoppt die Ausgabe; es besteht dann die Wahl zwischen Fortsetzung der Auflistung und Rücksprung ins Hauptmenü.

»Titelliste ausdrucken«: Durch diesen Unterpunkt gelangt man in das Druckmenü.

»Daten pflegen«: Hiermit wird das Datenpflege-Menü aufgerufen, in dem verschiedene Dienstprogramme enthalten sind.

»Programminformation«: Auf einer Info-Seite werden die aktuelle Versionsnummer und der Programmautor angezeigt. Ein Druck auf eine beliebige Taste bringt Sie ins Hauptmenü zurück.

»Startmenü«: Dieser Punkt bewirkt einen Rücksprung in das Startmenü.

Das Druckmenü(Bild 4)

»Titel alphabetisch drucken«: Alle in der Datei vorhandenen Titel werden in alphabetischer Reihenfolge auf dem Drucker ausgegeben.

»nach Kategorien ausdrucken«: Hier werden die einzelnen Titel nach der Reihenfolge der Kategorien ausgedruckt, wobei Start- und Endkategorie (von »A« bis »T«) anzugeben sind.

»nach Cass.Nummern ausdrucken«: Die Titel werden in der Reihenfolge der Kassettennummern ausgedruckt. Auch hier ist eine Vorwahl von Start- und Endnummer (Nummer 1 bis 9999) vorgesehen.

»nach Kriterien drucken«: Dieser Menüpunkt erlaubt eine sehr flexible Datenausgabe, da eine Auswahl nach individuellen Suchkriterien erfolgt. Es erscheint zunächst die Datensatzmaske für den Eintrag der Suchkriterien. Anschließend werden alle Datensätze, die den genannten Kriterien entsprechen, ausgedruckt.

»nach Datensatznummern drucken«: Jeder Datensatz erhält bei der Eingabe automatisch eine Nummer zugeteilt. Mit diesem Punkt lassen sich die Datensätze also in der Reihenfolge der Eingabe ausdrucken.

»Hauptmenü«: Dieser Menüpunkt bringt Sie zurück ins Hauptmenü.

Das Datenpflege-Menü (Bild 5)

»Index sortieren«: Da die Indexdatei nur bei Programmende, nicht aber bei der Dateneingabe automatisch sortiert wird, kann hier jederzeit während der Dateiarbeit ein Sortiervorgang durchgeführt werden.

»Index speichern«: Vor allem bei sehr vielen Ergänzungen oder Änderungen einer Datei ist es ratsam, nach einiger Zeit den Index auf der Diskette zu sichern. Dies geschieht über diesen Menüpunkt. Vor dem Abspeichern wird der Index noch sortiert; das obige Untermenü braucht daher nicht extra angewählt zu werden.

»Index wiederherstellen«: Wird das Programm nicht ordnungsgemäß verlassen, z.B. durch einen versehentlichen Reset des Computers oder durch einen plötzlichen Stromausfall, dann kann es passieren, daß der Index der Datei nicht mehr vollständig bzw. nicht ganz korrekt vorhanden ist, weil z.B. zwischenzeitlich erfolgte Neueinträge fehlen. Durch dieses Untermenü läßt sich jedoch der komplette Index rekonstruieren. Dieser Vorgang dauert, vor allem bei umfangreichen Dateien, einige Zeit.

»Datum ändern«: Das bei Programmbeginn erfragte Tagesdatum, das in der Eingabemaske und beim Listenausdruck erscheint, kann hier noch einmal geändert werden. Bei versehentlicher Anwahl dieses Punktes ist ein Verlassen mit der oberen Cursortaste (< ↑ >) möglich.

»Directory anzeigen«: Hiermit wird das Inhaltsverzeichnis der Programm- wie der Datendiskette auf dem Bildschirm angezeigt.

»Diskettenbefehl senden«: Zur Überprüfung der Einträge auf der Datendiskette kann hier der COLLECT-Befehl durchgeführt werden; außerdem ist es möglich, einzelne Files zu löschen (SCRATCH-Befehl). Einzelheiten siehe Benutzerhandbuch der Floppystation.

»Hauptmenü«: Dieser Menüpunkt bringt Sie zurück ins Hauptmenü.

Horizontale Menüs

Dies waren alle vertikalen Menüs des Programms. Auch die nachfolgend erläuterten horizontalen Menüs werden mit den *oberen* Cursortasten des C128 gesteuert: < → > bewirkt dabei ein schrittweises Springen des revers dargestellten Balkens nach rechts, < ← > bewegt ihn nach links. Die Auswahl der gewünschten Funktion erfolgt mit der

Bild 7. Das »Kategorien«-Fenster hilft dem Gedächtnis beim Anlegen neuer Datensätze auf die Sprünge

< ↑ >-Taste (siehe auch Hinweise auf der durch die HELP-Taste aufrufbare Hilfsseite). Analog dazu ist auch eine Steuerung mittels Joystick oder Maus am Control-Port 2 möglich, wobei zur Anwahl der Feuerknopf gedrückt werden muß.

Die Eingabemaske (Bild 6)

Die Eingabemaske erreichen Sie vom Hauptmenü aus über den Menüpunkt »Titel eingeben«. Es erscheinen alle Felder eines Datensatzes, wobei die Eingabefelder auf dem Bildschirm unterstrichen sind. Der Cursor innerhalb der einzelnen Felder wird hier durch die unteren Cursortasten zeichen- bzw. feldweise bewegt. Bei der Eingabe ist zu beachten, daß bei den Kategorien nur die jeweiligen Anfangsbuchstaben benötigt werden. Die Felder »Farbe«, »Dolby«, »Longplay«, »Stereo« und »Hifi« beantworten Sie bitte mit »j« oder »ja« bzw. »n« oder »nein«; nur der jeweils erste Buchstabe wird hier kontrolliert. Der Texteditor innerhalb der Felder entspricht dem vom Basic her gewohnten Modus, <INST> und <DEL> können also wie üblich benutzt werden.

Eingabe der Daten

Das horizontale Menü der Eingabemaske besteht aus fünf Punkten, die sich im wesentlichen selbst erklären: Mit »Hauptmenü« gelangt man in die vorherige Menüebene zurück, »Speichern« bewirkt ein Speichern des aktuellen Datensatzes auf die Datendiskette, bei »Speichern und Drucken« erfolgt zusätzlich eine Druckausgabe des Datensatzes. Sollten Ihnen bei der Eingabe die Kategorien nicht mehr geläufig sein, so wählen Sie den Menüpunkt »Kategorien« an. Dadurch werden alle Kategorien in einem Bildschirm-Fenster angezeigt (Bild 7). Wie bei einem vertikalen Menü kann daraus die entsprechende Filmart ausgesucht werden, wobei ein automatischer Übertrag in das Eingabefeld erfolgt, auch wenn der Cursor in einem anderen Feld steht. Der Menüpunkt »Statistik« informiert in einem weiteren Fenster über die Anzahl der Datensätze, die Belegung der Datei (in Prozent) und über die größte bisher vergebene Kassettennummer.

Menübereich »Titel suchen, ändern, löschen«

Auch hier erscheint wieder die Eingabemaske, allerdings sind die einzelnen Punkte des horizontalen Menüs etwas anders als bei »Titel eingeben«. Der Menüpunkt »Suchen« sucht innerhalb der Datei nach dem Eintrag, der den angegebenen Kriterien entspricht. Dabei werden die Suchkriterien ähnlich wie ein Neueintrag eingegeben. Es ist grundsätzlich nicht nötig, vollständige Angaben zu machen, beispielsweise genügt es, zum Auffinden eines Filmes nur einen Teil des Titels zu nennen; allerdings sollte man die Groß- und Kleinschreibung von Suchbegriffen beachten.

Suche nach allen Kriterien

Je mehr und je spezifischere Kriterien aufgeführt wurden, desto genauer ist die Suche. Eine Suche ist für alle Felder möglich, erheblich schneller ist der Suchvorgang jedoch, wenn Sie sich auf die drei Indexfelder (Titel, Kategorie und Kassettennummer) beschränken. Die Felder haben bei dieser Funktion eine unterschiedliche Wertigkeit, die (auf die Eingabemaske bezogen) von oben nach unten abnimmt. Es wird also zunächst nach dem Titel gesucht (sofern eingetragen), dann nach der Kategorie, der Kassettennummer etc.

Durch den Menüpunkt »Drucken« erfolgt ein Ausdruck des gefundenen und angezeigten Titels, mit »Weiter suchen« wird die Datei nach dem nächsten Titel, der den Suchkriterien entspricht, durchgesehen. Durch den Menüpunkt »Ändern« gelangt man in die Eingabemaske und kann den aktuellen Datensatz beliebig abändern und erneut speichern. Der Punkt »Löschen« bewirkt – nach einer zusätzlichen Sicherheitsabfrage – ein Löschen des Datensatzes im Index. Der Datensatz selbst wird nicht gelöscht, sondern beim nächsten Neueintrag einfach überschrieben. Bis dahin ist es möglich, diesen schon abgelegten Titel durch Anwählen der Funktion »Index wiederherstellen« im Dateipflege-Menü wieder in den aktuellen Bestand aufzunehmen.

Menübereich »Titelliste auf Monitor«

Auch hier existiert wieder ein horizontales Menü, das im wesentlichen den ersten drei Punkten des Druckmenüs entspricht: »Listen nach Alphabet«, »Listen nach Kategorien« und »Listen nach Kassettennummern«. In den letzten beiden Fällen ist wiederum eine Definition von Start- und Endnummern bzw. -kriterien vorgesehen.

Mit dieser Kurzanleitung und etwas Übung sollte ein fehlerfreies Bedienen des Programms ohne große Probleme möglich sein. Zu beachten ist vor allem die Unterscheidung der beiden Menüarten (vertikal und horizontal). Auf der Programmservice-Diskette zu diesem Sonderheft befindet sich zudem eine compilierte Fassung von »Videoarchiv«, mit der sich eine wesentlich höhere Arbeitsgeschwindigkeit ergibt.

(Detlef Erdmann/Dr. Rudolf Egg)

```
100 REM DELCO VIDEOARCHIV-SYSTEM                           <IIA>
102 :                                                      <2SB>
103 REM COMPILERANWEISUNGEN FUER AUSTRO-COMP 128           <IUB>
104 :                                                      <3CD>
105 REM **NG                                               <I6B>
106 REM **RI                                               <IA9>
108 REM **FI I,I1,I2,I3,J,J1,J2,T,SI,BI                    <JIE>
110 :                                                      <2SB>
120 FAST                                                   <PJ3>
130 COLOR 0,1: COLOR 4,1: PRINT "{CLR}" CHR$(14);          <B2U>
140 POKE 0,PEEK(0) OR 6                        : REM D
    EUTSCHE TASTATUR AUF                                   <0DG>
150 POKE 1,PEEK(1) AND 6                       : REM D
    AUER EINSCHALTEN                                       <HVH>
160 POKE 792,98                               : REM ST
    OP/RESTORE SPERREN                                     <UI5>
170 POKE 247,128                              : REM TA
    STENUMSCHALTUNG SPERREN                                <VOG>
180 GRAPHIC 5,1                               : REM 80
     ZEICHEN BILDSCHIRM                                    <N9A>
185 IF PEEK(215)=0 THEN SYS 4919                : REM
    UMSCHALTEN AUF 80-ZEICHEN                              <HRN>
190 :                                                      <6SR>
195 REM CENTRONICS-SOFT-INTERFACE LADEN                    <GE1>
200 REM DIE NAECHSTEN DREI ZEILEN NUR ABTIPPEN             <J2C>
201 REM WENN DRUCKER UEBER USERPORT ANGESTEUERT!           <J6C>
205 FI$="VIDEO-CENTRONICS": GOSUB 44000                    <D6I>
210 BLOAD(FI$),B0,P4864                                    <PAI>
215 SYS 486                                     : REM
     CENTRONICS-INTERFACE AKTIVIEREN                       <293>
220 :                                                      <0C1>
225 REM ZEICHENSATZ LADEN UND INITIALISIEREN               <J6C>
230 :                                                      <0S3>
240 FI$="VIDEO-CHARACTERS": GOSUB 44000                    <DKK>
250 BLOAD(FI$),B1,P53248                                   <UKL>
260 POKE 58,208: CLR                               : R
    EM START DER ZEROPAGE-KOPIE DER                        <KL3>
262 P1=PEEK(674): P2=PEEK(675): P3=PEEK(676)       : R
    EM KERNAL "FETVEC"-ROUTINE WIRD                        <458>
264 FOR I=0 TO 9: READ A: POKE 2816+I,A: NEXT
     : REM MIT "JSR 0B00" UEBERSCHRIEBEN                   <QED>
266 DATA 162,1,189,240,247,170,173,0,255,96 : REM
     CONFIGURATION-REGISTER MIT                            <4BK>
268 POKE 674,32: POKE 675,0: POKE 676,11
    : REM "BANK 1" LADEN, DA ZS. DORT                      <HLN>
270 BANK 15: SYS 5274                              : R
    EM SPRUNG IN ZS.-LADEROUTINE                           <02T>
272 POKE 674,P1: POKE 675,P2: POKE 676,P3
    : REM URSPRUENGLICHE "FETVEC"-ROUTINE                  <E52>
274 POKE 58,255: CLR                               : R
    EM WIEDERHERSTELLEN                                    <8QI>
280 :                                                      <5CL>
290 REM FUNKTIONSTASTEN MIT CHR$-CODE BELEGEN              <HA5>
300 :                                                      <0C1>
310 RESTORE 310: FOR I=1 TO 8: READ D: KEY I,CHR$
    (D): NEXT                                              <CND>
320 DATA 133,137,134,138,135,139,136,140                   <VKB>
330 POKE 4105,1: POKE 4123,132                 : REM "
    HELP" MIT CHR$(132) BELEGEN                            <6PG>
340 :                                                      <2C9>
490 DIM DT%(12),K$(19),C$(19),U$(9),PU$(114),PR$(
    255),RE%(1100),G$(1100),S%(30,2)                       <431>
500 FOR I=1 TO 12: READ D: DT%(I)=D: NEXT : REM K
    ALENDERDATEN                                           <FRI>
510 DATA 31,29,31,30,31,30,31,31,30,31,30,31               <6VK>
520 :                                                      <7CT>
530 BI=1: SI=1: GOSUB 11000: SCNCLR : BI=0: SI=0:
     GOSUB 11000: SCNCLR                                   <0ML>
535 POKE 2592,1                                : REM TA
    STATURPUFFER = 1                                       <P96>
536 :                                                      <6CP>
540 UM%=8: DM%=0: UD%=8: DD%=0                  : REM
    AUTOKONFIGURATOR                                       <GE0>
550 OPEN 1,UD%,15,"I1"                         : REM ST
    ELLT ART UND ZAHL DER                                  <FP9>
560 IF DS=0 THEN DD%=1: GOTO 59                  : REM
     ANGESCHLOSSENEN FLOPPIES FEST                         <BCJ>
570 CLOSE 1                                                <GAN>
575 TRAP 610                                               <GJD>
580 OPEN 1,9,15,"I0": UD%=9                                <U40>
590 CLOSE 1                                                <GMN>
600 GOTO 620                                               <PUA>
610 IF ER THEN UD%=8: RESUME 590                           <LPP>
620 TRAP                                                   <U7T>
630 :                                                      <4SJ>
880 TI%=0: GOSUB 2200                    : REM TITELBIL
    D AUFRUFEN                                             <JCQ>
885 BI=0: SI=1: GOSUB 11000: PRINT "{CTRL+G}";             <H44>
890 FL%=0                                                  <HIF>
900 Z1$="AEIJN "                         : REM JA/NEIN     <GBE>
910 Z2$="ABCDEFGHIJKLMNOPQRST"           : REM KATEGORIE
    N                                                      <PF4>
915 Z3$="0123456789"                     : REM ZIFFERN     <GE1>
920 Z4$=Z3$+Z2$+"UVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXY
    ZFCX±=-V@/=-+.,;:?!#'$%&()[]<>π*{SHF.SPACE}"
    : REM ERLAUBTE ZEICHEN                                 <MKE>
925 Z5$="{BROWN}++++++++++++++++++++++++++++++++++++
    ++++++++++++++++++++++++++++++++++++++++
    ++{GREEN}"                                             <RD5>
930 Z6$="{79SPACE}"                                        <H1I>
935 T1$(1)="FARBE , ": T1$(2)="S / W , "                   <KNN>
940 U$(5)="-------------------------------------
    ---------------------------------------"               <I85>
945 U$(6)="{15SPACE}D E L C O{2SPACE}-{2SPACE}V I
     D E O A R C H I V{2SPACE}-{2SPACE}S Y S T E
    M"                                                     <5VU>
950 U$(7)="CASS{3SPACE}FILMTITEL{49SPACE}LAUFZEIT
    {7SPACE}"                                              <D36>
955 U$(7)=U$(7)+"NR.{3SPACE}---------------------
    -----------------------------------------
    --------"                                              <4U4>
960 U$(7)=U$(7)+"{7SPACE}KATEGORIE{17SPACE}BEMERK
    UNGEN"                                                 <PML>
980 PUDEF "0"                                              <VLL>
990 IF JOY(2)>127 THEN 1000                                <JE8>
995 GET T$: IF T$="" THEN 990                              <RTQ>
997 :                                                      <F5S>
998 REM STARTMENUE                                         <LQN>
999 :                                                      <CLI>
1000 TE$(1)=" ARBEIT MIT BESTEHENDER DATEI "               <3R3>
1005 TE$(2)="{3RIGHT} DATENDISKETTE WECHSELN "             <OQ8>
1010 TE$(3)="{2RIGHT} ANLEGEN EINER NEUEN DATEI "          <RTT>
1020 TE$(4)="{2RIGHT} FESTGELEGTE DATEN FNDERN "           <1IP>
1030 TE[illegible])="{3RIGHT} VIDEO-ARCHIV VERLASSEN "     <TG4>
1040 TE$(9)="{3DOWN,CTRL+B,GREEN}S T A R T M E N
     X :{2DOWN}"                                           <4KF>
1045 GOSUB 22010                                           <0MJ>
1050 BY=11: TE=5: F3%=0: GOSUB 36000                       <Q3U>
1060 ON U% GOTO 3500,3800,2000,3000,4000                   <J3G>
1070 :                                                     <2GB>
1098 REM HAUPTMENUE                                        <0C1>
1099 :                                                     <70U>
1100 TE$(1)="{7RIGHT} TITEL EINGEBEN "                     <SH8>
1110 TE$(2)="TITEL SUCHEN, FNDERN , LCSCHEN"               <8U2>
1120 TE$(3)="{3RIGHT} TITELLISTE AUF MONITOR "             <NP4>
1130 TE$(4)="{3RIGHT} TITELLISTE AUSDRUCKEN "              <ED7>
1140 TE$(5)="{7RIGHT} DATEN PFLEGEN "                      <DA9>
1150 TE$(6)="{4RIGHT} PROGRAMMINFORMATION "                <L3B>
1160 TE$(7)="{5RIGHT} S T A R T M E N X "                  <KQ3>
1165 TE$(9)="{2DOWN,CTRL+B,GREEN}H A U P T M E N
     X :{2DOWN}"                                           <70S>
1170 IF DT$="" THEN F3%=0: GOSUB 9400                      <DAJ>
1180 GOSUB 22010                                           <C6J>
1190 BY=10: TE=7: GOSUB 36000                              <TGE>
1200 ON U% GOTO 5000,6000,7000,8000,1300,8800,100
     0                                                     <V2F>
```





```
1220 :                                                      <209>
1298 REM DATEIPFLEGE-MENUE                                  <1C5>
1299 :                                                      <50M>
1300 TE$(1)="{6RIGHT} INDEX SORTIEREN "                     <LG2>
1310 TE$(2)="{6RIGHT} INDEX SPEICHERN "                     <VDN>
1320 TE$(3)="{3RIGHT} INDEX WIEDERHERSTELLEN "              <07T>
1330 TE$(4)="{7RIGHT} DATUM{2SPACE}ANDERN "                 <VSK>
1340 TE$(5)="{5RIGHT} DIRECTORY ANZEIGEN "                  <QTE>
1350 TE$(6)="{3RIGHT} DISKETTENBEFEHL SENDEN "              <O0N>
1360 TE$(7)="{6RIGHT} H A U P T M E N U "                   <20P>
1370 TE$(9)="{2DOWN,CTRL+B,GREEN}DATEIPFLEGE - ME
     NU :{2DOWN}"                                           <KQR>
1380 GOSUB 22010                                            <86J>
1390 BY=10: TE=7: F3%=0: GOSUB 36000                        <6IS>
1400 ON U% GOTO 9000,9100,9300,9400,9600,9800,110
     0                                                      <ST7>
1410 :                                                      <5GN>
1998 REM NEUE DATEI ANLEGEN                                 <4SJ>
1999 :                                                      <EPQ>
2000 TE$(1)="{5RIGHT} STANDARDKATEGORIEN "                  <F54>
2100 TE$(1)="{5RIGHT} STANDARDKATEGORIEN "                  <D5C>
2110 TE$(2)="{5RIGHT} EIGENE{2SPACE}KATEGORIEN "            <PQD>
2120 GOSUB 22010                                            <A6J>
2130 PRINT TAB(28)TE$(7)                                    <56B>
2140 PRINT TAB(19)"WELCHE KATEGORIEN SOLLEN VERWE
     NDET WERDEN?"                                          <TKQ>
2150 BY=14: TE=2: GOSUB 36000                               <O0C>
2160 IF U%=1 THEN K%=1                                      <HE5>
2170 IF U%=2 THEN BEGIN                                     <QJ9>
2180 GOSUB 22010                                            <06J>
2190 PRINT TAB(28)TE$(7)                                    <F73>
2200 PRINT TAB(26)"EIGENE KATEGORIEN ERSTELLEN:"            <GD4>
2210 WINDOW 0,14,79,20                                      <B9G>
2220 SI=1: BI=FL%: GOSUB 11000                              <VT2>
2230 FOR I=0 TO 19                                          <SGH>
2240 CHAR 1,24,5,"{BROWN}KATEGORIE "+CHR$(I+193)+
     " : {GREEN}"                                           <MTK>
2250 CHAR 1,39,6: PRINT CHR$(142)"{BROWN}NNNNNNNN
     NNNNNNN{GREEN}"+CHR$(14);                              <E2H>
2260 CHAR 1,39,5: LT=15: Z$=Z4$: F3%=2: GOSUB 400
     00                                                     <DQJ>
2270 K$(I)=LEFT$(G$+Z6$,15): PRINT "{4DOWN}"                <OFP>
2280 NEXT : K%=0: PRINT "{CLR}": WINDOW 0,0,79,24           <TSO>
2290 BEND                                                   <DPU>
2300 TE$(1)="{6RIGHT} EPSON (PARALLEL) "                    <BHA>
2310 TE$(2)=" EPSON{2SPACE}(SERIELL + INTERFACE)
     "                                                      <S0E>
2320 TE$(3)=" COMMODORE{2SPACE}MPS{SHF.SPACE}801
     / VC 1525 "                                            <E1G>
2340 TE$(4)="{5RIGHT} COMMODORE{2SPACE}MPS{SHF.SP
     ACE}803 "                                              <L7C>
2350 GOSUB 22010                                            <96J>
2360 PRINT TAB(28)TE$(7)                                    <46F>
2370 PRINT TAB(22)"GEBEN SIE BITTE IHREN DRUCKERT
     YP AN:"                                                <A1L>
2380 BY=14: TE=4: GOSUB 36000                               <M9C>
2390 IF U%=1 THEN DR$="EPSON*": SA%=0: CW%=0                <8C5>
2392 IF U%=3 THEN DR$="MPS 801*": SA%=7: CW%=1              <LSG>
2396 IF U%=4 THEN DR$="MPS 803*": SA%=7: CW%=1              <LPG>
2400 IF U%=2 THEN BEGIN                                     <R7F>
2410 TE$(1)="{2RIGHT} GORLITZ VC-DIN-INTERFACE "            <U0B>
2420 TE$(2)="{3RIGHT} GORLITZ VCEI-INTERFACE "              <KNU>
2430 TE$(3)="{3RIGHT} DATA-BECKER{2SPACE}INTERFAC
     E "                                                    <KHD>
2440 GOSUB 22010                                            <66J>
2450 PRINT TAB(28)TE$(7)                                    <97R>
2460 PRINT TAB(20)"GEBEN SIE BITTE IHREN INTERFAC
     E-TYP AN:"                                             <MQ5>
2470 BY=14: TE=3: GOSUB 36000                               <P8C>
2480 IF U%=1 THEN SA%=7: CW%=0                              <L3B>
2490 IF U%=2 THEN SA%=12: CW%=1                             <CMA>
2500 IF U%=3 THEN SA%=1: CW%=1                              <LFB>
2510 DR$="EPSON*"                                           <4LA>
2520 BEND                                                   <PRE>
2530 GOSUB 22010                                            <36J>
2540 PRINT TAB(28)TE$(7)                                    <A7N>
2550 PRINT TAB(28)"BENUTZERDATEN FESTLEGEN:"                <UDO>
2560 WINDOW 0,14,79,20                                      <91P>
2570 SI=1: BI=FL%: GOSUB 11000                              <FB2>
2580 FOR I=1 TO 4                                           <UU0>
2590 CHAR 1,15,5: PRINT "{BROWN}ZEILE "I" : {GREE
     N}"                                                    <A4H>
2600 CHAR 1,27,6: PRINT CHR$(142)"{BROWN}NNNNNNNN
     NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN{GREEN}"+CHR
     $(14);                                                 <OJL>
2610 CHAR 1,27,5: LT=40: Z$=Z4$: F3%=2: GOSUB 400
     00                                                     <EEL>
2620 U$(I)=LEFT$(Z6$,(LT/2)-(LEN(G$)/2))+G$: U$(I
     )=U$(I)+LEFT$(Z6$,LT-(LEN(U$(I)))): PRINT "{
     4DOWN}"                                                <5C8>
2630 NEXT : PRINT "{CLR}": WINDOW 0,0,79,24                 <O6B>
2650 GOSUB 22010                                            <36J>
2660 PRINT TAB(28)TE$(7)                                    <E77>
2665 SI=1: BI=FL%: GOSUB 11000                              <N02>
2670 GOSUB 35000                                            <4E6>
2680 CHAR 1,0,22,Z5$: CHAR 1,0,24: PRINT "{21SPAC
     E}DIE BENOTIGTEN DATEIEN WERDEN GELADEN !{12
     SPACE}";                                               <FC1>
2690 IF K%=1 THEN BEGIN                                     <6F9>
2695 FI$="VIDEO-KATEGORIEN": GOSUB 45000: IF DR%<
     >0 THEN 2680                                           <AQT>
2700 DOPEN #1,(FI$),D(DM%),U(UM%)                           <90T>
2710 FOR I=0 TO 19                                          <KGH>
2720 INPUT#1,K$(I)                                          <LJN>
2730 NEXT                                                   <68G>
2740 DCLOSE #1 ON U(UM%)                                    <GBR>
2750 BEND                                                   <LQU>
2755 FI$="VIDEO-"+DR$: GOSUB 45000: IF DR%<>0 THE
     N 2680                                                 <PGK>
2760 DOPEN #1,(FI$),D(DM%),U(UM%)                           <50T>
2770 FOR I=32 TO 222                                        <CQQ>
2780 INPUT#1,PR$(I)                                         <AJB>
2790 NEXT                                                   <EDG>
2800 DCLOSE #1 ON U(UM%)                                    <MR1>
2805 IF F1%=1 THEN RETURN                                   <70T>
2810 GOSUB 35100                                            <UU4>
2820 CHAR 1,0,24: PRINT "{9SPACE}DIE DATENDISKETT
     E WIRD ANGELEGT. BITTE EINIGE MINUTEN GEDULD
      !";                                                   <HN4>
2830 FI$="VIDEOARCHIV DATA": HEADER(FI$),I01,D(DD
     %),U(UD%)                                              <SA7>
2835 IF DR%<>0 THEN GOSUB 45020: GOTO 2820                  <L9B>
2840 DOPEN #1,"VIDEO-DATEI{5SPACE}",L112,D(DD%),U
     (UD%)                                                  <J9F>
2850 RECORD #1,1101,1                                       <HS9>
2860 PRINT#1,CHR$(255);                                     <5H4>
2870 IF DS>1 AND DS<>50 THEN GOSUB 45020: GOTO 28
     20                                                     <BK3>
2880 DCLOSE #1 ON U(UD%)                                    <GBR>
2890 DOPEN #1,"VIDEO-STANDARDS ",D(DD%),U(UD%),W            <U97>
2910 FOR I=0 TO 19: PRINT#1,CHR$(34)+K$(I): NEXT            <VFO>
2920 PRINT#1,SA%: PRINT#1,CW%                               <D4C>
2930 FOR I=32 TO 222: PRINT#1,CHR$(34)+PR$(I): NE
     XT                                                     <MA9>
2940 FOR I=1 TO 4: PRINT#1,CHR$(34)+U$(I): NEXT             <V7T>
2950 DCLOSE #1 ON U(UD%)                                    <JJK>
2955 IF F1%=1 THEN RETURN                                   <4GJ>
2960 DOPEN #1,"VIDEO-INDEX{5SPACE}",D(DD%),U(UD%)
     ,W                                                     <KGH>
2970 PRINT#1,0                                              <8HR>
2980 DCLOSE #1 ON U(UD%)                                    <GRP>
2990 GOTO 1000                                              <40K>
2995 :                                                      <BPE>
2998 REM FESTGELEGTE DATEN AENDERN                          <75V>
2999 :                                                      <8P2>
3000 IF F2%=0 THEN GOSUB 22010: CHAR 1,0,22,Z5$:
     PRINT TAB(28)"{DOWN,CTRL+G}KEINE DATEN IM SP
     EICHER !";: SI=1: BI=FL%: GOSUB 11000: SLEEP
      3: GOTO 1000                                          <2DO>
3056 TE$(9)=""                                              <N34>
3060 TE$(7)="{2DOWN,CTRL+B,GREEN}FESTGELEGTE DATE
     N ANDERN:{2DOWN}"                                      <03B>
3070 F1%=1                                                  <QBR>
3080 GOSUB 2000                                             <H71>
3155 GOSUB 35100                                            <EE4>
3156 CHAR 1,0,22,Z5$                                        <I5K>
3157 CHAR 1,0,24,"{23SPACE}DIE DATENDISKETTE WIRD
      GEANDERT.{23SPACE}"                                   <D2U>
3158 FI$="VIDEO-STANDARDS ": GOSUB 45010: IF DR%<
     >0 THEN 3157                                           <562>
3160 SCRATCH(FI$),D(DD%),U(UD%)                             <1AP>
3170 GOSUB 2890                                             <HVO>
3180 F1%=0                                                  <I2R>
3190 GOTO 1000                                              <10K>
3200 :                                                      <70T>
3498 REM DATEIEN LADEN                                      <0C1>
3499 :                                                      <70U>
3500 IF F2%=1 THEN 110                      : REM IN
     DEX GELADEN ?                                          <BN3>
3510 GOSUB 22010                                            <56J>
3520 SI=1: BI=FL%: GOSUB 11000                              <V92>
3530 GOSUB 35100                                            <2U4>
3540 CHAR 1,0,22,Z5$: CHAR 1,0,24: PRINT "{21SPAC
     E}DIE BENOTIGTEN DATEIEN WERDEN GELADEN !{12
     SPACE}";                                               <1C1>
3545 FI$="VIDEO-STANDARDS ": GOSUB 45010: IF DR%<
     >0 THEN 1000                                           <95G>
3550 DOPEN #1,(FI$),D(DD%),U(UD%)                           <50F>
3570 FOR I=0 TO 19: INPUT#1,K$(I): NEXT                     <JLE>
3580 INPUT#1,SA%,CW%                                        <PF6>
3590 FOR I=32 TO 222: INPUT#1,PR$(I): NEXT                  <D48>
3600 FOR I=1 TO 4: INPUT#1,U$(I): NEXT                      <VD1>
3610 DCLOSE #1 ON U(UD%)                                    <K3A>
3615 FI$="VIDEO-INDEX{5SPACE}": GOSUB 45010: IF D
     R%<>0 THEN 1000                                        <M9F>
3620 DOPEN #1,(FI$),D(DD%),U(UD%)                           <70F>
3630 INPUT#1,N%                                             <OLO>
3640 IF N%=0 THEN 3660                                      <POA>
3650 FOR I=1 TO N%: INPUT#1,G$(I): INPUT#1,RE%(I)           <DJE>
3655 IF CA%<VAL(MID$(G$(I),17,4)) THEN CA%=VAL(MI
     D$(G$(I),17,4))                                        <40M>
3657 NEXT                                                   <898>
3660 DCLOSE #1 ON U(UD%)                                    <LRD>
3670 F2%=1: GOTO 1100                                       <N9G>
3680 :                                                      <70T>
3798 REM DATENDISKETTE WECHSELN                             <1S7>
3799 :                                                      <40I>
3800 IF F2%=0 THEN 3500                                     <9JB>
3810 F1%=1: GOSUB 9100: F1%=0                               <OS1>
3815 CA%=0: FOR I=1 TO N%: G$(I)="": RE%(I)=0: NE
     XT                                                     <FD0>
3820 CHAR 1,0,24: PRINT "{9SPACE}BITTE DIE {CTRL+
     O}NEUE"+CHR$(27)"O"+" DATENDISKETTE EINLEGEN
      UND EINE TASTE DRUCKEN !{8SPACE}";                     <C4M>
```

Listing 1. Das Hauptprogramm »VIDEOARCHIV«. Bitte mit dem Checksummer 128 (Seite 158) eingeben und vor dem Start unbedingt speichern!

```
3830 IF JOY(2)>127 THEN 3850                          <74L>
3840 GET TT$: IF TT$="" THEN 3830                     <7VR>
3850 F2%=0: GOTO 3540                                 <NTJ>
3997 :                                                <994>
3998 REM VIDEO-ARCHIV VERLASSEN                       <6SR>
3999 :                                                <APA>
4000 GOSUB 22010                                      <I6J>
4010 CHAR 1,0,22,Z5$                                  <3LI>
4020 PRINT TAB(11)"{DOWN}VIDEO-ARCHIV VERLASSEN?{
     4SPACE}SIND SIE SICHER ( JA / NEIN ) ?";         <5HS>
4030 SI=1: BI=FL%: GOSUB 11000: PRINT "{CTRL+G}";     <CHI>
4040 GET KEY T$                                       <600>
4050 IF T$="N" OR T$="N" THEN 1000                    <A5U>
4060 IF T$="J" OR T$="J" THEN BEGIN                   <AC8>
4065 IF F2%=0 THEN 4080                               <V3H>
4070 F1%=1: GOSUB 9100: F1%=0             : REM I
     NDEX SORTIEREN UND SPEICHERN                     <M0V>
4080 CHAR 1,0,24: PRINT "{11SPACE}DENKEN SIE AN I
     HRE DATENSICHERUNG !{6SPACE}AUF WIEDERSEHEN
     !{7SPACE}";                                      <7QI>
4090 SLEEP 3: SCNCLR : FL%=1-FL%: SI=0: BI=FL%: G
     OSUB 11000: SCNCLR                               <EUB>
4100 SYS 65341                                        <VVU>
4110 BEND                                             <LIU>
4120 GOTO 4040                                        <004>
4130 :                                                <BHF>
4998 REM TITEL EINGEBEN                               <1S7>
4999 :                                                <40I>
5000 FI$="VIDEO-DATEI{5SPACE}": GOSUB 45010: IF D
     R%<>0 THEN 1100                                  <TK1>
5002 DOPEN #2,(FI$),L112,D(DD%),U(UD%): TE$="E I
     N G A B E :"                                     <QIK>
5005 TE$(1)="{2SPACE}HAUPTMENX{2SPACE,RVOFF}": TE
     $(2)="{2SPACE}SPEICHERN{2SPACE,RVOFF}": TE$(
     3)="{2SPACE}SPEICHERN & DRUCKEN{2SPACE,RVOFF
     }": TE$(4)="{2SPACE}KATEGORIEN{2SPACE,RVOFF}
     ": TE$(5)="{2SPACE}STATISTIK{2SPACE,RVOFF}":
      TE=5: U%=2: F3%=1                               <8NC>
5010 F4%=1: FOR I=1 TO 10: ZL(I)=0: GG$(I)="": NE
     XT                                               <543>
5015 FL%=1-FL%: SI=0: BI=FL%: GOSUB 11000: GOSUB
     20000                                            <LRT>
5020 CHAR 1,0,22,Z5$                                  <JPJ>
5030 CHAR 1,0,24,+TE$(1)+"{RVSON}"+TE$(2)+"{RVOFF
     }"+TE$(3)+TE$(4)+TE$(5)                          <0BP>
5040 SI=1: BI=FL%: GOSUB 11000                        <U12>
5050 D=1: DO UNTIL RE%(D)=0 OR D=1100: D=D+1: LOO
     P                                                <TBL>
5080 IF D=1100 THEN BEGIN                             <IDJ>
5090 CHAR 1,0,24,"{CTRL+O} DATEI VOLL !"+CHR$(27)
     +"O"+"{2SPACE}BITTE NEUE DATENDISKETTE ANLEG
     EN ODER DIE DATEI REORGANISIEREN !"+CHR$(7)      <A0L>
5095 IF JOY(2)>127 THEN 5105                          <68G>
5100 GET T$: IF T$="" THEN 5095                       <1TK>
5105 GOTO 1100                                        <5CM>
5110 BEND                                             <5GU>
5120 CHAR 1,75,1: PRINT USING "####";D                <9KD>
5130 IF F4%<1 THEN F4%=10                             <L42>
5135 IF F4%=11 THEN F4%=1                             <TQF>
5140 IF F4%=1 THEN Z$=Z4$: LT=55: CX=15: CY=4         <FH5>
5150 IF F4%=2 THEN Z$=Z2$: LT=1: CX=15: CY=7          <EKI>
5160 IF F4%=3 THEN Z$=Z3$: LT=4: CX=15: CY=10         <I9L>
5170 IF F4%=4 THEN Z$=Z3$: LT=3: CX=36: CY=10         <2EH>
5180 IF F4%=5 THEN Z$=Z1$: LT=4: CX=15: CY=13         <R48>
5190 IF F4%=6 THEN Z$=Z1$: LT=4: CX=36: CY=13         <13C>
5200 IF F4%=7 THEN Z$=Z1$: LT=4: CX=15: CY=16         <NHI>
5210 IF F4%=8 THEN Z$=Z1$: LT=4: CX=36: CY=16         <1MM>
5220 IF F4%=9 THEN Z$=Z1$: LT=4: CX=53: CY=16         <BVQ>
5230 IF F4%=10 THEN Z$=Z4$: LT=42: CX=15: CY=19       <B9M>
5240 ZL=ZL(F4%): G$=GG$(F4%): WINDOW CX,CY,CX+LT+
     1,CY: PRINT TAB(ZL): GOSUB 40010                 <9TC>
5250 GOTO 5130                                        <105>
5260 :                                                <811>
5998 REM TITEL SUCHEN, AENDERN, LOESCHEN              <0S3>
5999 :                                                <60Q>
6000 FI$="VIDEO-DATEI{5SPACE}": GOSUB 45010: IF D
     R%<>0 THEN 1100                                  <HLH>
6002 DOPEN #2,(FI$),L112,D(DD%),U(UD%): TE$="{15S
     PACE}"                                           <28E>
6005 TE$(1)="{2SPACE}HAUPTMENX{2SPACE,RVOFF}": TE
     $(2)="{2SPACE}TITEL SUCHEN{2SPACE,RVOFF}": T
     E$(3)="{2SPACE}TITEL FNDERN{2SPACE,RVOFF}":
     TE$(4)="{2SPACE}TITEL LTSCHEN{2SPACE,RVOFF}"
     : TE=4: U%=1: F3%=1: D=0                         <40F>
6010 F4%=1: F6%=1: F7%=0: FOR I=1 TO 10: ZL(I)=0:
      GG$(I)="": NEXT                                 <3SR>
6015 FL%=1-FL%: SI=0: BI=FL%: GOSUB 11000: GOSUB
     20000                                            <DRT>
6020 CHAR 1,0,22,Z5$                                  <J1G>
6030 CHAR 1,0,24,"{RVSON}"+TE$(1)+"{RVOFF}"+TE$(2
     )+TE$(3)+TE$(4)                                  <753>
6040 SI=1: BI=FL%: GOSUB 11000                        <UP2>
6050 GOTO 5120                                        <3G4>
6100 IF U%=1 THEN 41120                               <A1A>
6105 IF GG$(1)="" AND GG$(2)="" AND GG$(3)="" AND
      GG$(4)="" AND GG$(5)="" AND GG$(6)="" AND G
     G$(7)="" AND GG$(8)="" AND GG$(9)="" AND GG$
     (10)="" THEN BEGIN                               <IUG>
6107 CHAR 1,0,24,"{24SPACE}KEIN SUCHKRITERIUM VOR
     HANDEN !{6SPACE}"+CHR$(7): SLEEP 3: GOTO 600
     5                                                <4LC>
6108 BEND                                             <1KE>
6109 TE$(1)="{2SPACE}HAUPTMENX{2SPACE,RVOFF}": TE
     $(2)="{2SPACE}WEITER SUCHEN{2SPACE,RVOFF}":
     TE$(3)="{2SPACE}DRUCKEN{2SPACE,RVOFF}": TE$(
     4)="{2SPACE}+NDERN{2SPACE,RVOFF}": TE$(5)="{
     2SPACE}LTSCHEN{2SPACE,RVOFF,7SPACE}": TE=5:
     IF U%>2 THEN U%=U%+1                             <TI0>
6120 FOR I=1 TO N%                                    <LR8>
6125 IF G$(I)="" THEN 6525                            <SGF>
6130 IF LEN(GG$(1))>0 THEN BEGIN                      <HAE>
6132 IF LEN(GG$(1))>15 THEN IF INSTR(GG$(1),LEFT$
     (G$(I),15))=0 THEN 6525                          <MC9>
6133 IF LEN(GG$(1))<=15 THEN IF INSTR(LEFT$(G$(I)
     ,15),GG$(1))=0 THEN 6525                         <HOP>
6134 BEND                                             <LMU>
6135 IF LEN(GG$(2))>0 THEN IF INSTR(MID$(G$(I),16
     ,1),GG$(2))=0 THEN 6525                          <9PM>
6140 IF LEN(GG$(3))>0 THEN IF INSTR(MID$(G$(I),17
     ,4),GG$(3))=0 THEN 6525                          <6DG>
6310 D=VAL(RIGHT$(G$(I),4))                           <PID>
6315 RECORD #2,D,1: IF DS<>0 THEN GOSUB 45020: GO
     TO 1100                                          <MB3>
6320 INPUT#2,D$                                       <61S>
6325 FOR J=5 TO 9                                     <T36>
6330 IF LEFT$(GG$(J),1)="J" THEN GG$(J)="1"           <7P5>
6335 IF LEFT$(GG$(J),1)="N" THEN GG$(J)="2"           <29J>
6340 NEXT                                             <Q70>
6360 IF LEN(GG$(4))>0 THEN IF INSTR(MID$(D$,61,3)
     ,GG$(4))=0 THEN 6525                             <0M8>
6365 IF LEN(GG$(5))>0 THEN IF INSTR(MID$(D$,64,1)
     ,GG$(5))=0 THEN 6525                             <64G>
6370 IF LEN(GG$(6))>0 THEN IF INSTR(MID$(D$,65,1)
     ,GG$(6))=0 THEN 6525                             <NG0>
6375 IF LEN(GG$(7))>0 THEN IF INSTR(MID$(D$,66,1)
     ,GG$(7))=0 THEN 6525                             <B1G>
6380 IF LEN(GG$(8))>0 THEN IF INSTR(MID$(D$,67,1)
     ,GG$(8))=0 THEN 6525                             <CD3>
6385 IF LEN(GG$(9))>0 THEN IF INSTR(MID$(D$,68,1)
     ,GG$(9))=0 THEN 6525                             <0QJ>
6390 IF LEN(GG$(10))>0 THEN IF INSTR(RIGHT$(D$,42
     ),GG$(10))=0 THEN 6525                           <J81>
6393 IF F6%=3 THEN F7%=1: GOSUB 8560: GOTO 6525       <I0D>
6395 F7%=1: CHAR 1,0,24: C$(U%)="{RVSON}": FOR J=
     1 TO TE: PRINT C$(J)TE$(J);: NEXT                <U0L>
6400 CHAR 1,75,1: PRINT USING "####";D                <9GD>
6405 CHAR 1,15,4,LEFT$(D$,55): CHAR 1,15,7,MID$(D
     $,56,1): CHAR 1,20,7,K$(ASC(MID$(D$,56,1))-6
     5)                                               <QGN>
6410 CHAR 1,15,10,MID$(D$,57,4): CHAR 1,36,10,MID
     $(D$,61,3)                                       <600>
6415 IF MID$(D$,64,1)="0" THEN CHAR 1,15,13,"{4SP
     ACE}"                                            <5TE>
6420 IF MID$(D$,64,1)="1" THEN CHAR 1,15,13,"JA{2
     SPACE}"                                          <0HQ>
6425 IF MID$(D$,64,1)="2" THEN CHAR 1,15,13,"NEIN
     "                                                <IRM>
6430 IF MID$(D$,65,1)="0" THEN CHAR 1,36,13,"{4SP
     ACE}"                                            <25E>
6435 IF MID$(D$,65,1)="1" THEN CHAR 1,36,13,"JA{2
     SPACE}"                                          <6PQ>
6440 IF MID$(D$,65,1)="2" THEN CHAR 1,36,13,"NEIN
     "                                                <N3M>
6445 IF MID$(D$,66,1)="0" THEN CHAR 1,15,16,"{4SP
     ACE}"                                            <V7E>
6450 IF MID$(D$,66,1)="1" THEN CHAR 1,15,16,"JA{2
     SPACE}"                                          <RBQ>
6455 IF MID$(D$,66,1)="2" THEN CHAR 1,15,16,"NEIN
     "                                                <91M>
6460 IF MID$(D$,67,1)="0" THEN CHAR 1,36,16,"{4SP
     ACE}"                                            <OVE>
6465 IF MID$(D$,67,1)="1" THEN CHAR 1,36,16,"JA{2
     SPACE}"                                          <S3Q>
6470 IF MID$(D$,67,1)="2" THEN CHAR 1,36,16,"NEIN
     "                                                <EPM>
6475 IF MID$(D$,68,1)="0" THEN CHAR 1,53,16,"{4SP
     ACE}"                                            <1JE>
6480 IF MID$(D$,68,1)="1" THEN CHAR 1,53.16."JA{2
     SPACE}"                                          <2VQ>
6485 IF MID$(D$,68,1)="2" THEN CHAR 1,53,16,"NEIN
     "                                                <GLM>
6490 CHAR 1,15,19,RIGHT$(D$,42)                       <03C>
6495 GOSUB 36510                                      <T61>
6500 IF U%=1 THEN 6005                                <4KJ>
6505 IF U%=2 THEN 6525                                <ESR>
6510 IF U%=3 THEN CHAR 1,0,24,"{26SPACE}DATENSATZ
      WIRD GEDRUCKT !{11SPACE}": GOSUB 33500: U%=
     2: GOTO 6495                                     <R1R>
6515 IF U%=4 THEN 6800                                <DG1>
6520 IF U%=5 THEN 6900                                <043>
6525 NEXT                                             <D38>
6530 IF F7%=0 THEN BEGIN                              <4T2>
6532 CHAR 1,0,24,"{7SPACE}KEIN TITEL MIT DEN ANGE
     GEBENEN KRITERIEN IN DER DATEI VORHANDEN !{5
     SPACE}"+CHR$(7): SLEEP 3: IF F6%=3 THEN 8005
     : ELSE GOTO 6005                                 <0LJ>
6533 BEND                                             <RJ6>
6534 IF F6%=3 THEN 8650                               <7Q3>
6535 TE$(2)="{RVOFF,22SPACE}KEIN WEITERER TITEL I
     N DER DATEI VORHANDEN !"+CHR$(7): TE=1: U%=1     <QC2>
6540 F7%=0: CHAR 1,0,24,"{RVSON}"+TE$(1)+TE$(2):
     GOSUB 36510: GOTO 6005                           <9DI>
6545 :                                                <994>
6798 REM GELADENEN DATENSATZ WEITERVERARBEITEN        <4SJ>
6799 :                                                <EPQ>
6800 TE$(1)="{2SPACE}HAUPTMENX{2SPACE,RVOFF}": TE
     $(2)="{2SPACE}SPEICHERN{2SPACE,RVOFF}": TE$(
     3)="{2SPACE}SPEICHERN & DRUCKEN{2SPACE,RVOFF
     }": TE$(4)="{2SPACE}KATEGORIEN{2SPACE,RVOFF}
     ": TE$(5)="{2SPACE}STATISTIK{2SPACE,RVOFF}"      <FD4>
6803 TE=5: U%=2: F3%=1: F4%=1: F6%=2                  <EHP>
6805 GG$(1)=LEFT$(D$,55): II=55: DO UNTIL MID$(GG
```

```
     $(1),II,1)<>" ": II=II-1: LOOP : GG$(1)=LEFT
     $(GG$(1),II)                   : REM "SPACES"
     ENTFERNEN                                       <NLF>
6810 GG$(2)=MID$(D$,56,1): GG$(3)=MID$(D$,57,4):
     GG$(4)=MID$(D$,61,3)                            <E5K>
6815 FOR J=5 TO 9                                    <NJ6>
6820 IF MID$(D$,59+J,1)="1" THEN GG$(J)="JA"         <AL9>
6825 IF MID$(D$,59+J,1)="2" THEN GG$(J)="NEIN"       <DDP>
6830 IF MID$(D$,59+J,1)="0" THEN GG$(J)=""           <BNE>
6835 NEXT                                            <KEO>
6840 GG$(10)=RIGHT$(D$,42): II=42: DO UNTIL II=1
     OR MID$(GG$(10),II,1)<>" ": II=II-1: LOOP       <09D>
6845 IF II=0 THEN GG$(10)="": ELSE GG$(10)=LEFT$(
     GG$(10),II)                                     <B8S>
6850 FOR J=1 TO 10: ZL(J)=0: NEXT                    <08F>
6855 CHAR 1,0,24,TE$(1)+"{RVSON}"+TE$(2)+TE$(3)+T
     E$(4)+TE$(5)                                    <Q3G>
6860 GOTO 5120                                       <784>
6865 IF U%=1 THEN 6005                               <D8K>
6870 IF U%=2 OR U%=3 THEN 34500                      <UN9>
6875 IF U%=4 THEN 32000                              <KKL>
6880 IF U%=5 THEN 31000                              <68V>
6885 :                                               <288>
6898 REM TITEL LOESCHEN                              <3CD>
6899 :                                               <106>
6900 CHAR 1,0,24,"{9SPACE}SOLL DER DATENSATZ WIRK
     LICH GELTSCHT WERDEN ?{2SPACE}( JA / NEIN ){
     8SPACE}"+CHR$(7)                                <MQF>
6910 GET KEY T$                                      <10D>
6920 IF T$="N" OR T$="N" THEN 6495                   <AH1>
6930 IF T$="J" OR T$="J" THEN RE%(D)=0: G$(I)=""     <U4V>
6940 CHAR 1.0.24.Z6$                                 <UG6>
6950 CHAR 1,0,24,"{20SPACE}DATENSATZ NUMMER": PRI
     NT D;" WURDE GELTSCHT !{14SPACE}";: SLEEP 3:
      GOTO 6495                                      <NH6>
6960 :                                               <50L>
6998 REM TITELLISTE AUF BILDSCHIRM                   <3SF>
6999 :                                               <002>
7000 FI$="VIDEO-DATEI{5SPACE}": GOSUB 45010: IF D
     R%<>0 THEN 1100                                 <LL1>
7005 DOPEN #2,(FI$),L112,D(DD%),U(UD%)               <PJ7>
7010 TE$(1)="{2SPACE}HAUPTMENX{2SPACE,RVOFF,3SPAC
     E,BROWN}LISTEN NACH:{GREEN,2SPACE}": TE$(2)=
     "{2SPACE}ALPHABET{3SPACE,RVOFF}": TE$(3)="{2
     SPACE}KATEGORIEN{2SPACE,RVOFF}": TE$(4)="{2S
     PACE}CASS.NUMMERN{2SPACE,RVOFF}": TE=4: U%=2
     : F5%=1                                         <5OM>
7030 FL%=1-FL%: SI=0: BI=FL%: GOSUB 11000            <65C>
7040 PRINT "{CLR,BROWN,5SPACE}T FILMTITEL{47SPACE
     }T LAUFZEIT"                                    <Q9T>
7050 PRINT "CASS.T++++++++++++++++++£+++++++£+++++
     ++++++++++++++++++++++++++T++++++++++++"        <IK1>
7070 PRINT "NR.{2SPACE}T KATEGORIE{7SPACE}T{7SPAC
     E}T BEMERKUNGEN"                                <LV5>
7080 PRINT "+++++T++++++++++++++++++T+++++++T+++++
     +++++++++++++++++++++++++++++++++++++++"        <7E8>
7090 CHAR 1,0,22,Z5$                                 <3TJ>
7100 GOSUB 36510                                     <OM1>
7110 IF U%=1 THEN WINDOW 0,0,79,24: DCLOSE #2 ON
     U(UD%): GOTO 1100                               <NKM>
7120 IF U%=2 THEN 8100                               <A18>
7130 IF U%=3 THEN 8200                               <RLE>
7140 IF U%=4 THEN 8300                               <BLD>
7500 IF M%=9 THEN TE$(1)="{2SPACE}HAUPTMENX{2SPAC
     E,RVOFF}": TE$(2)="{2SPACE}NFCHSTE SEITE{2SP
     ACE,RVOFF,48SPACE}": TE=2: U%=2: GOTO 7510      <LM9>
7505 IF M%>1 THEN TE$(1)="{RVSON,2SPACE}HAUPTMENX
     {2SPACE,RVOFF}": TE$(2)="{RVOFF,31SPACE}KEIN
     E WEITEREN EINTRFGE VORHANDEN !"+CHR$(7): TE
     =1: U%=1                                        <NT6>
7510 CHAR 1,0,24,TE$(1)+"{RVSON}"+TE$(2)             <ONM>
7520 WINDOW 0,4,79,20: CHAR 1,0,16                   <NLR>
7530 FOR I2=1 TO M%-1                                <L43>
7540 D$=MID$(PU$(I2),57,4)+"{3SPACE}"+LEFT$(PU$(I
     2),55)+" , "+MID$(PU$(I2),61,3)+" MINUTEN{12
     SPACE}"                                         <2A7>
7550 D$=D$+"{BROWN}"+K$(ASC(MID$(PU$(I2),56,1))-6
     5)+" , "+T1$(VAL(MID$(PU$(I2),64,1)))+RIGHT$
     (PU$(I2),42)+"{GREEN}"                          <LIM>
7560 PRINT D$                                        <RLE>
7570 NEXT I2                                         <CBG>
7580 WINDOW 0,0,79,24                                <P9A>
7590 GOSUB 36510                                     <PM1>
7600 IF U%=1 THEN 7010                               <5CK>
7610 IF U%=2 THEN RETURN                             <1K0>
7620 :                                               <A19>
7998 REM TITELLISTE DRUCKEN                          <25B>
7999 :                                               <20A>
8000 FI$="VIDEO-DATEI{5SPACE}": GOSUB 45010: IF D
     R%<>0 THEN 1100                                 <9IH>
8002 DOPEN #2,(FI$),L112,D(DD%),U(UD%)               <MEO>
8005 TE$(1)="{RIGHT} TITEL ALPHABETISCH DRUCKEN "    <7G8>
8010 TE$(2)="{RIGHT} NACH KATEGORIEN AUSDRUCKEN "    <TO7>
8020 TE$(3)=" NACH CASS.NUMMERN AUSDRUCKEN "         <ABT>
8025 TE$(4)="{3RIGHT} NACH KRITERIEN DRUCKEN "       <N9I>
8030 TE$(5)=" NACH DATENSATZNUMMERN DRUCKEN "        <MRF>
8040 TE$(6)="{6RIGHT} H A U P T M E N X "            <6SP>
8050 TE$(9)="{2DOWN,CTRL+B,GREEN}TITELLISTE DRUCK
     EN :{2DOWN}"                                    <BOT>
8060 GOSUB 22010                                     <42J>
8070 F5%=0: F6%=0: F8%=0: BY=10: TE=6: GOSUB 3600
     0                                               <D9G>
8071 IF U%=6 THEN U%=1: GOTO 41100                   <I0I>
8075 GOSUB 22010: CHAR 1,0,22,Z5$                    <87Q>
8085 ON U% GOTO 8100,8200,8300,8500,8400             <VNS>
8090 :                                               <NIV>
8098 REM TITEL ALPHABETISCH DRUCKEN                  <9D5>
8099 :                                               <LQM>
8100 GOSUB 8700: SI=1: BI=FL%: GOSUB 11000           <JIM>
8110 M%=1: FOR I=1 TO N%                             <OHL>
8120 D=VAL(RIGHT$(G$(I),4))                          <ACD>
8125 IF RE%(D)=0 THEN 8160                           <7AI>
8130 RECORD #2,(D),1: IF DS<>0 THEN GOSUB 45020:
     GOTO 1100                                       <HES>
8135 INPUT#2,D$                                   :
      REM DATENSATZ LADEN                            <K74>
8140 PU$(M%)=D$: M%=M%+1                             <Q8U>
8150 IF F5%=0 THEN IF M%=F8%+109 THEN GOSUB 33000
     : M%=1      : REM DRUCKROUTINE                  <G0M>
8155 IF F5%=1 THEN IF M%=9 THEN GOSUB 7500: M%=1     <T6B>
8160 NEXT                                            <2O0>
8162 IF M%>1 THEN IF F5%=0 THEN GOSUB 33000          <H1H>
8163 IF M%>1 THEN IF F5%=1 THEN GOSUB 7500           <0G8>
8165 IF F5%=1 THEN 7010                              <V7D>
8170 GOTO 8005                                       <05I>
8180 :                                               <M2P>
8198 REM NACH KATEGORIEN SUCHEN                      <9T7>
8199 :                                               <KQI>
8200 CHAR 1,12,9,"GEBEN SIE BITTE DIE ZU ": IF F5
     %=1 THEN PRINT "LISTENDEN";: ELSE PRINT "DRU
     CKENDEN";                                       <L2D>
8201 PRINT " KATEGORIEN AN{2SPACE}(A-T) :": IF F5
     %=1 THEN F3%=6: ELSE F3%=5                      <O5A>
8202 CHAR 1,31,15,"{BROWN}ERSTE KATEGORIE: {GREEN
     ,2SPACE}": CHAR 1,48,16,CHR$(142)+"{BROWN}N{
     GREEN}"+CHR$(14)                                <SP7>
8203 CHAR 1,30,17,"{BROWN}LETZTE KATEGORIE: {GREE
     N,2SPACE}": CHAR 1,48,18,CHR$(142)+"{BROWN}N
     {GREEN}"+CHR$(14)                               <RRC>
8204 CHAR 1,48,15: LT=1: Z$=MID$(Z4$,11,20): SI=1
     : BI=FL%: GOSUB 11000: GOSUB 40000              <NVD>
8205 J1=ASC(G$)                                      <F6K>
8206 CHAR 1,48,17: GOSUB 40000                       <OAM>
8207 J2=ASC(G$)                                      <EAK>
8208 IF J1>J2 THEN 8202                              <DQM>
8209 WINDOW 0,5,79,20: PRINT "{CLR}": WINDOW 0,0,
     79,24                                           <8U7>
8210 GOSUB 8700                                      <867>
8229 M%=1: FOR J=J1 TO J2: FOR I=1 TO N%             <Q9J>
8230 IF INSTR(MID$(G$(I),16,1),CHR$(J))=1 THEN BE
     GIN                                             <J3P>
8235 D=VAL(RIGHT$(G$(I),4))                          <I7D>
8240 RECORD #2,(D),1: IF DS<>0 THEN GOSUB 45020:
     GOTO 1100                                       <PES>
8245 INPUT#2,D$                                      <OLS>
8250 PU$(M%)=D$: M%=M%+1                             <T82>
8260 IF F5%=0 THEN IF M%=F8%+109 THEN GOSUB 33000
     : M%=1                                          <EK5>
8265 IF F5%=1 THEN IF M%=9 THEN GOSUB 7500: M%=1     <N73>
8267 BEND                                            <VBM>
8270 NEXT : NEXT                                     <VOH>
8272 IF M%>1 THEN IF F5%=0 THEN GOSUB 33000: GOTO
      8275                                           <F6R>
8273 IF M%>1 THEN IF F5%=1 THEN GOSUB 7500: GOTO
     8275                                            <D54>
8274 IF M%=1 THEN CHAR 1,0,24,"{13SPACE}KEIN EINT
     RAG IN DEN GEWFHLTEN KATEGORIEN VORHANDEN !{
     12SPACE}"+CHR$(7): SLEEP 3                      <Q2F>
8275 IF F5%=1 THEN 7010                              <VRE>
8280 GOTO 8005                                       <2DI>
8290 :                                               <LIN>
8298 REM NACH CASSETTENNUMMERN SUCHEN                <8D1>
8299 :                                               <NQU>
8300 CHAR 1,11,9,"GEBEN SIE BITTE DIE ZU ": IF F5
     %=1 THEN PRINT "LISTENDEN";: ELSE PRINT "DRU
     CKENDEN";                                       <NEE>
8301 PRINT " CASSETTEN-NUMMERN AN :": IF F5%=1 TH
     EN F3%=6: ELSE F3%=5                            <3KP>
8302 CHAR 1,28,15,"{BROWN}ERSTE CASS.NUMMER: {GRE
     EN,5SPACE}": CHAR 1,48,16,CHR$(142)+"{BROWN}
     NNNN{GREEN}"+CHR$(14)                           <TGD>
8303 CHAR 1,27,17,"{BROWN}LETZTE CASS.NUMMER: {GR
     EEN,5SPACE}": CHAR 1,48,18,CHR$(142)+"{BROWN
     }NNNN{GREEN}"+CHR$(14)                          <4MI>
8304 CHAR 1,48,15: LT=4: Z$=Z3$: SI=1: BI=FL%: GO
     SUB 11000: GOSUB 40000                          <2A2>
8305 J1=VAL(G$): IF J1<1 THEN 8302                   <71P>
8306 CHAR 1,48,17: GOSUB 40000                       <OEM>
8307 J2=VAL(G$): IF J2<1 THEN 8302                   <M7P>
8308 IF J1>J2 THEN 8302                              <D2L>
8309 WINDOW 0,5,79,20: PRINT "{CLR}": WINDOW 0,0,
     79,24                                           <8E7>
8310 GOSUB 8700                                      <967>
8320 M%=1: FOR J=J1 TO J2: FOR I=1 TO N%             <N9J>
8330 IF VAL(MID$(G$(I),17,4))=J THEN BEGIN           <SIB>
8335 D=VAL(RIGHT$(G$(I),4))                          <I3D>
8340 RECORD #2,(D),1: IF DS<>0 THEN GOSUB 45020:
     GOTO 1100                                       <9ES>
8345 INPUT#2,D$                                      <QLS>
8350 PU$(M%)=D$: M%=M%+1                             <V8A>
8360 IF F5%=0 THEN IF M%=F8%+109 THEN GOSUB 33000
     : M%=1                                          <CK5>
8365 IF F5%=1 THEN IF M%=9 THEN GOSUB 7500: M%=1     <J7J>
8367 BEND                                            <NAM>
8370 NEXT : NEXT                                     <UOH>
8372 IF M%>1 THEN IF F5%=0 THEN GOSUB 33000: GOTO
      8375                                           <F6P>
8373 IF M%>1 THEN IF F5%=1 THEN GOSUB 7500: GOTO
     8375                                            <DT6>
```

Listing 1. (Fortsetzung)

```
8374 IF M%=1 THEN CHAR 1,0,24,"{16SPACE}DIESE CAS
     SETTENNUMMERN SIND NICHT VORHANDEN !{16SPACE
     }"+CHR$(7): SLEEP 3                                  <BOD>
8375 IF F5%=1 THEN 7010                                   <VJF>
8380 GOTO 8005                                            <2TI>
8390 :                                                    <KIJ>
8398 REM NACH DATENSATZNUMMERN DRUCKEN                    <8T3>
8399 :                                                    <MQQ>
8400 CHAR 1,7,9,"GEBEN SIE BITTE DIE ZU DRUCKENDE
     N DATENSATZ-NUMMERN AN{2SPACE}(1 -": PRINT N
     %"):": F3%=5                                         <MLJ>
8402 CHAR 1,25,15,"{BROWN}ERSTE DATENSATZ-NUMMER:
      {GREEN,5SPACE}": CHAR 1,49,16,CHR$(142)+"{B
     ROWN}NNNN{GREEN}"+CHR$(14)                           <4HP>
8403 CHAR 1,24,17,"{BROWN}LETZTE DATENSATZ-NUMMER
     : {GREEN,5SPACE}": CHAR 1,49,18,CHR$(142)+"{
     BROWN}NNNN{GREEN}"+CHR$(14)                          <0GG>
8404 CHAR 1,49,15: LT=4: Z$=Z3$: SI=1: BI=FL%: GO
     SUB 11000: GOSUB 40000                               <523>
8405 J1=VAL(G$): IF J1<1 OR J1>N% OR J1>1100 THEN
      8402                                                <RF6>
8406 CHAR 1,49,17: GOSUB 40000                            <GIM>
8407 J2=VAL(G$): IF J2<1 OR J2>N% OR J2>1100 THEN
      8402                                                <09U>
8408 IF J1>J2 THEN 8402                                   <FAS>
8409 WINDOW 0,5,79,20: PRINT "{CLR}": WINDOW 0,0,
     79,24                                                <BU7>
8410 GOSUB 8700                                           <E67>
8420 M%=1: FOR J=J1 TO J2: FOR I=1 TO N%                  <B9J>
8430 IF VAL(RIGHT$(G$(I),4))=J THEN BEGIN                 <84E>
8435 D=VAL(RIGHT$(G$(I),4))                               <IVD>
8440 RECORD #2,(D),1: IF DS<>0 THEN GOSUB 45020:
     GOTO 1100                                            <P2T>
8445 INPUT#2,D$                                           <KLS>
8450 PU$(M%)=D$: M%=M%+1                                  <H9I>
8460 IF M%=F8%+109 THEN GOSUB 33000: M%=1                 <D6T>
8465 BEND                                                 <BD6>
8470 NEXT : NEXT                                          <POH>
8475 IF M%>1 THEN GOSUB 33000                             <97S>
8480 GOTO 8005                                            <1DI>
8490 :                                                    <JIF>
8498 REM NACH KRITERIEN DRUCKEN                           <BDD>
8499 :                                                    <HQ6>
8500 TE$=" KRITERIENWAHL ": TE$(1)="{2SPACE}DRUCK
     MENY{2SPACE,RVOFF}": TE$(2)="{2SPACE}KRITERI
     ENAUSWAHL BEENDET{2SPACE,RVOFF}": TE=2: U%=2         <G4P>
8505 D=0: F3%=1: F4%=1: F6%=3: F7%=0                      <HQQ>
8507 FOR I=1 TO 10: ZL(I)=0: GG$(I)="": NEXT              <5I4>
8510 GOSUB 11000: GOSUB 20000                             <NJ6>
8515 CHAR 1,0,22,Z5$: CHAR 1,0,24,TE$(1)+"{RVSON}
     "+TE$(2)                                             <NNP>
8520 SI=1: BI=FL%: GOSUB 11000                            <TH2>
8525 GOTO 5120                                            <EC4>
8530 IF U%=1 THEN F6%=0: GOTO 8005                        <0HH>
8535 IF GG$(1)="" AND GG$(2)="" AND GG$(3)="" AND
      GG$(4)="" AND GG$(5)="" AND GG$(6)="" AND G
     G$(7)="" AND GG$(8)="" AND GG$(9)="" AND GG$
     (10)="" THEN BEGIN                                   <BDM>
8540 CHAR 1,0,24,"{24SPACE}KEIN SUCHKRITERIUM VOR
     HANDEN !{6SPACE}"+CHR$(7): SLEEP 3: F4%=1: G
     OTO 8515                                             <6RK>
8545 BEND                                                 <BD6>
8547 GOSUB 22010: GOSUB 8700: SI=1: BI=FL%: GOSUB
      11000                                               <1E6>
8550 M%=1: GOTO 6120                                      <ISU>
8560 PU$(M%)=D$: M%=M%+1                                  <G9M>
8565 IF M%=F8%+109 THEN GOSUB 33000: M%=1                 <T6T>
8570 RETURN                                               <Q67>
8650 IF M%>1 THEN GOSUB 33000                             <R5K>
8655 GOTO 8005                                            <3HI>
8670 :                                                    <MIR>
8698 REM DRUCKMELDUNG AUSGEBEN                            <AD9>
8699 :                                                    <JQE>
8700 CHAR 1,0,22,Z5$                                      <G9T>
8710 PRINT "{DOWN} DATENSFTZE WERDEN GEPUFFERT UN
     D ANSCHLIEVEND ";: IF F5%=0 THEN PRINT "GEDR
     UCKT.";: ELSE PRINT "GELISTET.";                     <Q1U>
8720 PRINT "{2SPACE}BITTE ETWAS GEDULD !";                <Q2A>
8730 RETURN                                               <Q67>
8740 :                                                    <M2P>
8798 REM PROGRAMMINFORMATION                              <ATB>
8799 :                                                    <IQA>
8800 GOSUB 22010                                          <I2J>
8810 PRINT CHR$(142)"{BROWN,5DOWN}OGGGGGGGGGGGGGG
     GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
     GGGGGGGGGGGGGGGGGGGGI";                              <ITP>
8820 FOR J=1 TO 6: PRINT "E" SPC(78)"E";: NEXT            <62C>
8830 PRINT "PHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
     HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHJ"
      CHR$(14);                                           <INQ>
8840 CHAR 1,35,12,"{BROWN}VERSION 2.21 D"                 <140>
8850 CHAR 1,17,13,"PROGRAMMIERT VON DETLEF ERDMAN
     N, 3502 VELLMAR"                                     <RL8>
8860 CHAR 1,28,15,CHR$(142)+"C"+CHR$(14)+" 1987 B
     Y MARKT &{SHF.SPACE}TECHNIK"                         <8N1>
8870 SI=1: BI=FL%: GOSUB 11000                            <CF2>
8880 IF JOY(2)>127 THEN 8900                              <HEE>
8890 GET T$: IF T$="" THEN 8880                           <HFJ>
8900 GOTO 1100                                            <COM>
8910 :                                                    <QJB>
8998 REM INDEX SORTIEREN                                  <DTN>
8999 :                                                    <SRI>
9000 GOSUB 22010                                          <62J>
9010 CHAR 1,0,22,Z5$                                      <0TV>
9020 PRINT TAB(12)"{DOWN}INDEXDATEI WIRD SORTIERT
      !{3SPACE}BITTE EINEN MOMENT GEDULD !";              <39H>
9030 SI=1: BI=FL%: GOSUB 11000                            <DF2>
9040 UF%=1: OF%=N%                                        <5RP>
9050 GOSUB 38000                                          <BBI>
9060 GOTO 1300                                            <9OI>
9070 :                                                    <IIB>
9098 REM INDEX SICHERN                                    <8D1>
9099 :                                                    <NQU>
9100 GOSUB 22010                                          <42J>
9105 SI=1: BI=FL%: GOSUB 11000                            <L82>
9110 GOSUB 35100                                          <0Q4>
9120 CHAR 1,0,22,Z5$                                      <GTV>
9130 PRINT TAB(4)"{DOWN}INDEXDATEI WIRD SORTIERT
     UND GESPEICHERT!{4SPACE}BITTE EINEN MOMENT G
     EDULD !";                                            <4B5>
9140 UF%=1: OF%=N%                                        <DRP>
9150 GOSUB 38000                                          <9BI>
9152 FI$="VIDEO-INDEX{5SPACE}": GOSUB 45010: IF D
     R%<>0 THEN 9120                                      <VS9>
9155 SCRATCH(FI$),D(DD%),U(UD%)                           <8UP>
9160 DOPEN #1,"VIDEO-INDEX{5SPACE}",D(DD%),U(UD%)
     ,W                                                   <BKH>
9165 PRINT#1,N%                                           <LGQ>
9170 FOR I=1 TO N%                                        <CV8>
9180 PRINT#1,CHR$(34)+G$(I): PRINT#1,RE%(I)               <2RR>
9190 NEXT                                                 <ETG>
9200 DCLOSE #1 ON U(UD%)                                  <SAB>
9210 IF F1%=1 THEN RETURN                                 <QRA>
9220 GOTO 1300                                            <9OI>
9230 :                                                    <IIB>
9298 REM INDEX WIEDERHERSTELLEN                           <9D5>
9299 :                                                    <LQM>
9300 GOSUB 22010                                          <02J>
9305 SI=1: BI=FL%: GOSUB 11000                            <L02>
9308 GOSUB 35100                                          <1Q4>
9310 CHAR 1,0,24: PRINT "{11SPACE}INDEXDATEI WIRD
      WIEDERHERGESTELLT !{3SPACE}BITTE ETWAS GEDU
     LD !{7SPACE}";                                       <QQ3>
9315 FI$="VIDEO-DATEI{5SPACE}": GOSUB 45010: IF D
     R%<>0 THEN 9310                                      <1BI>
9320 N%=0                                                 <N6N>
9330 FOR I=1 TO 1100: G$(I)="": RE%(I)=0: NEXT            <NMU>
9340 DOPEN #2,(FI$),L112,D(DD%),U(UD%)                    <NMV>
9345 FOR I=1 TO 1100                                      <QQT>
9347 RECORD #2,(I),1: IF DS<>0 THEN GOSUB 45020:
     GOTO 1300                                            <IQ2>
9350 GET #2,D$                                            <J3U>
9351 IF D$=CHR$(255) OR D$="" THEN 9370                   <M5K>
9353 RECORD #2,(I),1: IF DS<>0 THEN GOSUB 45020:
     GOTO 1300                                            <AQ2>
9354 INPUT#2,D$                                           <V5S>
9355 IF LEN(D$)<110 THEN 9370                             <ACM>
9356 G$=RIGHT$("0000"+RIGHT$(STR$(I),LEN(STR$(I))
     -1),4)                                               <T2J>
9360 G$(N%+1)=LEFT$(D$,15)+MID$(D$,56,5)+G$               <11N>
9365 N%=N%+1: RE%(I)=1                                    <AUR>
9370 NEXT                                                 <6OG>
9375 DCLOSE #2 ON U(UD%)                                  <TQC>
9380 GOTO 9120                                            <29K>
9390 :                                                    <MIR>
9398 REM DATUM AENDERN                                    <9T7>
9399 :                                                    <KQI>
9400 GOSUB 22010                                          <E2J>
9410 IF U%=4 THEN F3%=3: PRINT TAB(33)"{2DOWN,CTR
     L+B,GREEN}DATUM FNDERN :{2DOWN}": ELSE PRINT
      "{4DOWN}"                                           <HN5>
9420 CHAR 1,13,15,"{BROWN}GEBEN SIE BITTE DAS DAT
     UM EIN{2SPACE}( TTMMJJJJ ) :{2SPACE,GREEN,BS
     PACE}"                                               <EOQ>
9430 CHAR 1,60,16: PRINT CHR$(142)"{BROWN}NNNNNNN
     N{GREEN}" CHR$(14)                                   <C8D>
9440 CHAR 1,60,15: LT=8: Z$=Z3$                           <3UV>
9450 SI=1: BI=FL%: GOSUB 11000                            <DJ2>
9460 GOSUB 40000                                          <6BA>
9470 IF VAL(LEFT$(G$,2))<1 OR VAL(LEFT$(G$,2))>31
      THEN 9420                                           <3UV>
9480 IF VAL(MID$(G$,3,2))<1 OR VAL(MID$(G$,3,2))>
     12 THEN 9420                                         <KF9>
9490 IF VAL(LEFT$(G$,2))>DT%(VAL(MID$(G$,3,2))) T
     HEN 9420                                             <BFT>
9500 IF VAL(LEFT$(G$,2))=29 AND VAL(MID$(G$,3,2))
     =2 THEN IF VAL(RIGHT$(G$,4))/4<>INT(VAL(RIGH
     T$(G$,4))/4) THEN 9420                               <0L7>
9505 IF VAL(RIGHT$(G$,4))<1986 THEN 9420                  <D49>
9510 DT$=LEFT$(G$,2)+"."+MID$(G$,3,2)+"."+RIGHT$(
     G$,4)                                                <K5S>
9520 IF U%<>4 THEN RETURN                                 <77J>
9530 GOTO 1300                                            <A0I>
9540 :                                                    <M2P>
9598 REM DIRECTORY ANZEIGEN                               <ATB>
9599 :                                                    <IQA>
9600 GOSUB 22010                                          <A2J>
9610 TE$(1)="{3SPACE}MENY{3SPACE,RVOFF}": TE$(2)=
     " DIR{SHF.SPACE}PROGRAMMDISK {RVOFF}": TE$(3
     )=" DIR DATENDISK {RVOFF}": TE$(4)=" FREI PR
     OGRAMMDISK {RVOFF}": TE$(5)=" FREI DATENDISK
      {RVOFF}"                                            <AOG>
9620 WINDOW 0,0,79,24                                     <VPG>
9630 CHAR 1,0,22,Z5$                                      <01S>
9640 TE=5: GOSUB 36500                                    <V0V>
9665 IF U%=1 THEN 1300                                    <9C4>
9670 SLOW                                                 <IQ0>
9680 IF U%=2 THEN GOSUB 35000: GOSUB 9730: DIRECT
     ORY D(DM%),U(UM%)                                    <H8P>
9690 IF U%=3 THEN GOSUB 35100: GOSUB 9730: DIRECT
     ORY D(DD%),U(UD%)                                    <EHF>
9700 IF U%=4 THEN GOSUB 35000: GOSUB 9730: DIRECT
```

```
     ORY "{SHF.SPACE}",D(DM%),U(UM%) : REM CHR$(1
     60)                                          <F09>
9710 IF U%=5 THEN GOSUB 35100: GOSUB 9730: DIRECT
     ORY "{SHF.SPACE}",D(DD%),U(UD%)              <JCS>
9720 FAST : IF DS<>0 THEN GOSUB 45020             <3C6>
9725 GOTO 9620                                    <1TQ>
9730 WINDOW 26,6,60,20                            <B9O>
9740 PRINT "{CLR}": CHAR 1,0,14                   <15Q>
9750 RETURN                                       <Q27>
9760 :                                            <L2L>
9798 REM DISKETTENBEFEHL SENDEN                   <BTF>
9799 :                                            <GQ2>
9800 GOSUB 22010                                  <M2J>
9810 CHAR 1,0,22,Z5$                              <1TR>
9820 TE$(1)="{4SPACE}MENÜ{4SPACE,RVOFF}": TE$(2)=
     "{2SPACE}COLLECT DATENDISK{2SPACE,RVOFF}": T
     E$(3)="{2SPACE}SCRATCH FILE AUF DATENDISK{2S
     PACE,RVOFF}"                                 <MB7>
9830 TE=3: F3%=0: GOSUB 36500                     <LAM>
9860 IF U%=1 THEN 1300                            <0G3>
9870 IF U%=2 THEN BEGIN                           <TBM>
9880 GOSUB 35100                                  <RQ4>
9890 COLLECT D(DD%),U(UD%)                        <DCB>
9900 IF DR%<>0 THEN GOSUB 45020: GOTO 9830        <BSH>
9910 BEND                                         <50U>
9920 IF U%=3 THEN BEGIN                           <TVL>
9930 WINDOW 0,5,79,20                             <KGT>
9940 CHAR 1,14,11,"{BROWN}NAME DES ZU LÖSCHENDEN
     EINTRAGES :{2SPACE,GREEN,16SPACE}"           <ULS>
9950 CHAR 1,50,12: PRINT CHR$(142)"{BROWN}NNNNNNN
     NNNNNNNNN{GREEN}" CHR$(14)                   <687>
9960 CHAR 1,50,11: LT=16: Z$=Z4$                  <BEF>
9970 F3%=4: GOSUB 40000                           <GM5>
9980 PRINT "{CLR}": WINDOW 0,0,79,24              <EAF>
9990 GOSUB 35100                                  <OQ4>
10000 SCRATCH(G$),D(DD%),U(UD%)                   <50K>
10010 IF DS>1 THEN GOSUB 45020: GOTO 9830         <Q13>
10020 BEND                                        <NUE>
10030 GOTO 9810                                   <E54>
10050 :                                           <58N>
10998 REM BILDSCHIMWECHSELROUTINE                 <7GV>
10999 :                                           <812>
11000 BANK 15: IF BI=0 THEN BI=0: AT=8: ELSE BI=1
      6: AT=24                                    <3M2>
11010 POKE 2606,BI: POKE 2607,AT: IF SI=1 THEN SY
      S 52684,BI,12: SYS 52684,AT,20              <JTF>
11020 RETURN                                      <G07>
11030 :                                           <48J>
19998 REM EINGABEMASKE                            <EHR>
19999 :                                           <Q3A>
20000 PRINT CHR$(142)"{CLR,GREEN}DIASN VXVI" CHR$
      (14)" VIDEO- {BROWN}K{13SPACE,GREEN}"TE$"{1
      3SPACE,BROWN}K DATUM: {GREEN}"DT$           <KCO>
20030 PRINT CHR$(142)"{GREEN}FJBTFUWUWJ" CHR$(14)
      " ARCHIV {BROWN}K{41SPACE}K DATENSATZ:{3SPA
      CE,GREEN}0000"                              <QF3>
20040 PRINT "{BROWN}++++++++++++++++++T++++++++++
      +++++++++++++++++++++++++++T+++++++++++
      +++++++"                                    <566>
20050 PRINT "{DOWN,6SPACE,BROWN}TITEL:{GREEN}     <5MU>
20060 PRINT CHR$(142)"{BROWN,15SPACE}NNNNNNNNNNNN
      NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
      " CHR$(14)                                  <K5A>
20070 PRINT                                       <VMU>
20080 PRINT "{2SPACE}KATEGORIE:{GREEN,7SPACE,BROW
      N}({GREEN,15SPACE,BROWN}) {GREEN}"          <BFT>
20090 PRINT CHR$(142)"{BROWN,15SPACE}N{3SPACE}NNN
      NNNNNNNNNNNNNN" CHR$(14)                    <594>
20100 PRINT                                       <NJU>
20110 PRINT "{3SPACE}CASS.NR.:{GREEN,12SPACE,BROW
      N}LAUFZEIT:{GREEN,8SPACE,BROWN}MINUTEN"     <6TS>
20120 PRINT CHR$(142)"{BROWN,15SPACE}NNNN{17SPACE
      }NNN" CHR$(14)                              <RDJ>
20130 PRINT                                       <FGU>
20140 PRINT "{6SPACE}FARBE:{GREEN,15SPACE,BROWN}D
      OLBY:{GREEN}"                               <K0V>
20150 PRINT CHR$(142)"{BROWN,15SPACE}NNNN{17SPACE
      }NNNN" CHR$(14)                             <Q4E>
20160 PRINT                                       <7LU>
20170 PRINT "{3SPACE}LONGPLAY:{GREEN,14SPACE,BROW
      N}STEREO:{GREEN,12SPACE,BROWN}HIFI:{GREEN}" <3PO>
20180 PRINT CHR$(142)"{BROWN,15SPACE}NNNN{17SPACE
      }NNNN{13SPACE}NNNN" CHR$(14)                <IDK>
20190 PRINT                                       <VQU>
20200 PRINT "BEMERKUNGEN:{GREEN}"                 <HEF>
20210 PRINT CHR$(142)"{BROWN,15SPACE}NNNNNNNNNNNN
      NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN" CHR$(14)    <E8G>
20220 RETURN                                      <HG7>
21130 :                                           <997>
21998 REM TITELBILD                               <0G3>
21999 :                                           <60Q>
22000 PRINT "{CLR,2DOWN}";: GOTO 22020            <9C7>
22010 FL%=1-FL%: SI=0: BI=FL%: GOSUB 11000: TI%=1
      : PRINT "{CLR}";                            <IEQ>
22020 PRINT TAB(9) CHR$(142)"{RED}OGGGGGGGGGGGG
      GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
      GGI"                                        <9IN>
22030 PRINT TAB(9)"E" SPC(22)"{YELLOW}DGIDGLN{2SP
      ACE}OGLOGI" SPC(22)"{RED}E"                 <QRK>
22040 PRINT TAB(9)"E" CHR$(14)"{YELLOW} VIDEOARCH
      IV-SYSTEM - " CHR$(142)"E EKQ E{2SPACE}E{2S
      PACE}E E " CHR$(14)"- VIDEOARCHIV-SYSTEM "
      CHR$(142)"{RED}E"                           <QHR>
22050 PRINT TAB(9)"E" SPC(22)"{YELLOW}FHJRHMFHMPH
      MPHJ" SPC(22)"{RED}E"                       <P9T>
22060 PRINT TAB(9)"PHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
      HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHJ{GREEN}" CHR
      $(14)                                       <3AR>
22065 IF TI%=1 THEN RETURN                        <LKK>
22070 PRINT TAB(11)"{2DOWN,GREEN}DIE INDEX-SEQUEN
      TIELLE VIDEOCASSETTENVERWALTUNG FÜR IHREN"  <DC4>
22080 PRINT TAB(21)"{DOWN,PURPLE}C O M M O D O R
      E{3SPACE}P C{3SPACE}1 2 8 ( D )"            <H61>
22090 PRINT TAB(27)"{2DOWN,YELLOW}EIN QUALITÄTSPR
      OGRAMM VON"                                 <3SI>
22100 PRINT TAB(34) CHR$(142)"{DOWN,WHITE}DIASN V
      XVI"                                        <U9D>
22110 PRINT TAB(34)"FJBTFUWUWJ"                   <A5U>
22120 PRINT TAB(31) CHR$(14)"{DOWN,YELLOW}S O F T
       W A R E"                                   <P9D>
22130 PRINT TAB(26) CHR$(142)"{4DOWN,GREEN}C " CH
      R$(14)"1987 BY MARKT & TECHNIK";            <7ON>
22140 RETURN                                      <H07>
22150 :                                           <C9J>
22998 REM HILFSSEITE                              <3GF>
22999 :                                           <002>
23000 GOSUB 22010                                 <VML>
23010 PRINT CHR$(27)"M";CHR$(142)"{BROWN,DOWN}OGG
      GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
      GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGI";          <BS9>
23020 FOR J=1 TO 17: PRINT "E" SPC(78)"E";: NEXT  <KSK>
23030 PRINT "PHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
      HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
      J" CHR$(14)                                 <V95>
23040 CHAR 1,2,7,"{RVSON,GREEN}MENÜSTEUERUNG:{BRO
      WN,RVOFF} DIE STEUERUNG ERFOLGT MITTELS "+C
      HR$(142)+"E"+CHR$(14)+" {RVSON,GREEN}EINGAB
      E:{BROWN,RVOFF} DIE CURSORSTEUERUNG"        <VMO>
23050 CHAR 1,17,8,"DER OBEREN CURSORSTEUERTASTEN
      "+CHR$(142)+"E"+CHR$(14)+"{2SPACE}ERFOLGT M
      ITTELS DER UNTEREN"                         <BOJ>
23055 CHAR 1,47,9,CHR$(142)+"E"+CHR$(14)+"{11SPAC
      E}CURSORSTEUERTASTEN"                       <NMJ>
23060 CHAR 1,2,10,"{CTRL+B}VERTIKALE MENÜS"+CHR$(
      27)+"O"+"{12SPACE,CTRL+B}HORIZONTALE MENÜS"
      +CHR$(27)+"O "+CHR$(142)+"E"                <SAO>
23065 CHAR 1,47,11,"E"+CHR$(14)+" {RVSON,GREEN}DE
      L:{BROWN,RVOFF,2SPACE}LÖSCHT 1 ZEICHEN LINK
      S"                                          <L20>
23070 CHAR 1,6,12,"HOCH": PRINT SPC(23)"WÄHLEN{8S
      PACE}"+CHR$(142)+"E"+CHR$(14)+"{19SPACE}VOM
       CURSOR"                                    <95R>
23080 CHAR 1,7,13,CHR$(142)+"{GREEN}-↑": PRINT SP
      C(26)"-↑{10SPACE,BROWN}E"                   <J03>
23090 CHAR 1,7,14,"{GREEN}]↑": PRINT SPC(26)"]↑{1
      0SPACE,BROWN}E"+CHR$(14)+" {RVSON,GREEN}INS
      T:{BROWN,RVOFF,3SPACE}FÜGT EIN LEERZEICHEN"
      +CHR$(142)                                  <KF5>
23100 CHAR 1,5,15,"{BROWN}YZ{2SPACE,GREEN}[£{BROW
      N} "+CHR$(14)+"WÄHLEN{9SPACE}LINKS"+CHR$(14
      2)+" {GREEN}YZ{2SPACE}[£ "+CHR$(14)+"{BROWN
      }RECHTS "+CHR$(142)+"E "+CHR$(14)+"{6SPACE}
      RECHTS VORM CURSOR EIN"                     <PG6>
23110 CHAR 1,7,16,CHR$(142)+"{GREEN}!"+CHR$(34):
      PRINT : CHAR 1,35,16,"{BROWN}!"+CHR$(34)+"{
      10SPACE}E"                                  <RK6>
23120 PRINT : CHAR 1,7,17,"{GREEN}#$": PRINT SPC(
      26)"{BROWN}#${10SPACE}E "+CHR$(14)+"{RVSON,
      GREEN}CURSOR DOWN:{BROWN,RVOFF} 1 FELD VORW
      ÄRTS"                                       <7LD>
23130 CHAR 1,5,18,"RUNTER": PRINT SPC(36) CHR$(14
      2)"E"                                       <NOS>
23140 CHAR 1,1,19,"GGGGGGGGGGGGGGGGGGGGGGGGGGG
      GGGGGGGGGGGGGGGE "+CHR$(14)+"{RVSON,GREEN}
      CURSOR UP:{BROWN,RVOFF,3SPACE}EIN FELD ZURÜ
      CK"                                         <9AR>
23145 CHAR 1,2,20,"{RVSON,GREEN}HELP:{BROWN,RVOFF
      } AUFRUF DIESER HILFSSEITE{15SPACE}"+CHR$(1
      42)+"E"                                     <22J>
23150 CHAR 1,1,21,"GGGGGGGGGGGGGGGGGGGGGGGGGGG
      GGGGGGGGGGGGGGGE "+ CHR$(14)+"{RVSON,GREEN
      }CURSOR LEFT/RIGHT:{BROWN,RVOFF,2SPACE}BEWE
      GUNG"                                       <PLN>
23160 CHAR 1,2,22,"DRUCKERTREIBER: ": IF SA%=0 AN
      D CW%=0 THEN PRINT "PARALLEL (CENTRONICS)":
       ELSE PRINT "SERIELL (COMMODORE-BUS)"       <20M>
23165 CHAR 1,47,22,CHR$(142)+"E"+CHR$(14)+"{4SPAC
      E}DES CURSORS INNERHALB DES"                <F1D>
23170 CHAR 1,2,23,"DATENDISKETTE : UNIT": PRINT U
      D%", DRIVE"DD%"{12SPACE}" CHR$(142)"E" CHR$
      (14)"{6SPACE}AKTUELLEN EINGABEFELDES"       <UIL>
23180 SI=1: BI=FL%: GOSUB 11000                   <6D2>
23190 IF JOY(2)>127 THEN 23210                    <EOF>
23200 GET T$: IF T$="" THEN 23190                 <N6O>
23210 PRINT CHR$(27)"L";: GOSUB 22010: GOTO 36020 <ENJ>
23220 :                                           <EFF>
30998 REM STATISTIK                               <3GF>
30999 :                                           <002>
31000 CHAR 1,60,6,"{BROWN}£+++++++++++++++++N"    <DA8>
31010 FOR J=1 TO 11: CHAR 1,60,6+J,"K{17SPACE}K":
       NEXT                                       <6HR>
31020 CHAR 1,60,18,"I+++++++++++++++++O"          <2BI>
31030 WINDOW 61,7,77,17: PRINT CHR$(27)"M";       <I9L>
31040 PRINT "{GREEN}S T A T I S T I K{BROWN}+++++
      ++++++++++++{GREEN}DATENSÄTZE:{2SPACE}1100
      (MAXIMAL)"                                  <I6I>
31050 PRINT "DATENSÄTZE:{2SPACE}";: PRINT USING "
      ####";N%: PRINT " (ISTWERT){7SPACE,BROWN}++
      +++++++++++++++"                            <PVM>
```

Listing 1. (Fortsetzung)

```
31060 PUDEF " "                                          <BT4>
31070 PRINT "{GREEN}BELEGUNG:";: PRINT USING "###
      .##";N%/11;: PRINT " %{BROWN}++++++++++++++
      +++";                                          <3T3>
31080 PUDEF "0"                                      <CDK>
31090 PRINT "{GREEN}GRCVTE CASSETTEN-NUMMER:{6SPA
      CE}";: PRINT USING "####";CA%                  <LFK>
31100 IF JOY(2)>127 THEN 31120                       <7PS>
31110 GET T$: IF T$="" THEN 31100                    <50H>
31120 WINDOW 60,6,78,18: PRINT "{CLR}";CHR$(27)"L
      ";: WINDOW 0,0,79,24                           <LOQ>
31130 GOTO 5120                                      <3S5>
31140 :                                              <EPP>
31998 REM KATEGORIEN ZEIGEN                          <2GB>
31999 :                                              <20A>
32000 CHAR 1,60,6,"{BROWN}E+++++++++++++++++N"      <BAG>
32010 FOR J=1 TO 11: CHAR 1,60,6+J,"K{17SPACE}K":
       NEXT                                          <AGB>
32020 CHAR 1,60,18,"I+++++++++++++++++C{GREEN}"     <3RO>
32030 WINDOW 61,7,77,17: PRINT CHR$(27)"M";          <U85>
32040 J1=0                                           <FTF>
32050 K%=0                                           <0SJ>
32060 C$(K%)="{RVSON}"                               <PMF>
32070 PRINT "{HOME}";: FOR J=J1 TO J1+10: PRINT C
      $(J)" "K$(J)" {RVOFF}";: NEXT                  <0K7>
32080 IF JOY(2)<>0 THEN 32100                        <FSE>
32090 GET T$: IF T$="" THEN 32080                    <4SN>
32100 T=PEEK(212): C$(K%)=""                         <SHJ>
32110 IF T=86 OR JOY(2)>127 THEN 32150               <4TV>
32120 IF T=84 OR JOY(2)=4 OR JOY(2)=5 OR JOY(2)=6
       THEN BEGIN                                    <95D>
32122 IF K%>9 THEN J1=J1+1                           <CEC>
32124 IF J1>9 THEN J1=9                              <M8A>
32125 IF K%=19 THEN 32140                            <M2R>
32126 K%=1-(K%<19)*K%                                <SDM>
32128 BEND                                           <FHE>
32130 IF T=83 OR JOY(2)=1 OR JOY(2)=2 OR JOY(2)=8
       THEN BEGIN                                    <1R4>
32132 IF K%<10 THEN J1=J1-1                          <RIP>
32134 IF J1<0 THEN J1=0                              <6OE>
32135 IF K%=0 THEN 32140                             <RUS>
32136 K%=K%-1-19*(K%=0)                              <6G8>
32138 BEND                                           <BHU>
32140 GOTO 32060                                     <BPC>
32150 WINDOW 60,6,78,18: PRINT "{CLR}" CHR$(27)"L
      ";: FOR I3=1 TO 500: NEXT                      <SA8>
32160 WINDOW 0,0,79,24: CHAR 1,15,7,CHR$(K%+65):
      CHAR 1,20,7,K$(K%): GG$(2)=CHR$(K%+65)         <0JA>
32170 GOTO 5130                                      <1K4>
32180 :                                              <EPP>
32998 REM DRUCKROUTINE                               <1G7>
32999 :                                              <40I>
33000 TRAP 33700                                     <ELG>
33005 IF SA%=0 THEN OPEN 4,5: GOTO 33030             <9T5>
33010 OPEN 4,4,SA%                                   <FPD>
33030 IF F8%=6 THEN 33176                            <5MQ>
33070 PU$=U$(5): GOSUB 39000                         <L86>
33080 PRINT#4,PU$: PRINT#4                           <O1D>
33090 PU$=U$(6): GOSUB 39000                         <BF6>
33100 PRINT#4,PU$: PRINT#4                           <C3T>
33110 PU$=U$(5): GOSUB 39000                         <5A6>
33120 PRINT#4,PU$: PRINT#4                           <83D>
33125 IF F8%=1 THEN 33505                            <7ML>
33130 FOR I3=1 TO 4: PU$=U$(I3): GOSUB 39000         <1GK>
33140 PRINT#4,LEFT$(Z6$,20)+PU$: NEXT                <GAB>
33150 PRINT#4: PRINT#4,LEFT$(Z6$,31)+"STAND : "DT
      $: PRINT#4                                     <QQF>
33160 PU$=U$(5): GOSUB 39000                         <P9M>
33170 PRINT#4,PU$: PRINT#4: PRINT#4: PRINT#4         <206>
33175 I3=13                                          <1P6>
33176 MM%=1: IF M%-1<I3 THEN I3=M%-1                 <7BC>
33180 PU$=U$(5): GOSUB 39000                         <1EM>
33190 PRINT#4,PU$                                    <EQF>
33200 PU$=U$(7): GOSUB 39000                         <T9M>
33210 PRINT#4,PU$                                    <IQF>
33220 PU$=U$(5): GOSUB 39000                         <H8M>
33230 PRINT#4,PU$: PRINT#4                           <62L>
33240 FOR I2=1 TO I3                                 <KQE>
33250 PU$=MID$(PU$(MM%),57,4)+"{3SPACE}"+LEFT$(PU
      $(MM%),55)+" , "+MID$(PU$(MM%),61,3)+" MINU
      TEN{11SPACE}"                                  <EN1>
33260 PU$=PU$+K$(ASC(MID$(PU$(MM%),56,1))-65)+" ,
       "+T1$(VAL(MID$(PU$(MM%),64,1)))+RIGHT$(PU$
      (MM%),42): GOSUB 39000                         <7H1>
33270 PRINT#4,PU$: PRINT#4: MM%=MM%+1                <VL0>
33280 IF I3=19 THEN IF M%=MM% THEN 33350             <4I5>
33290 NEXT                                           <G6G>
33300 IF I3=13 OR I3=M%-1 THEN BEGIN                 <LCF>
33310 FOR I1=1 TO 72-20-I2*3: PRINT#4: NEXT          <T3N>
33320 IF M%=MM% THEN CLOSE 4: TRAP : RETURN          <D0N>
33330 I3=19: GOTO 33180                              <QH1>
33340 BEND                                           <7GE>
33350 IF I3=19 THEN BEGIN                            <DD4>
33360 IF M%=MM% THEN FOR I1=1 TO 72-6-I2*3: PRINT
      #4: NEXT : CLOSE 4: F8%=6: TRAP : RETURN       <UNT>
33370 FOR I1=1 TO 72-3-I2*3: PRINT#4: NEXT           <1F0>
33380 GOTO 33180                                     <FP4>
33390 BEND                                           <JMU>
33500 F8%=1: GOTO 33000                              <LHD>
33505 PRINT#4                                        <1O1>
33510 PU$="TITEL{7SPACE}: "+LEFT$(D$,55): : GOSUB
       39000: GOSUB 33640                            <ER9>
33520 PU$="KATEGORIE{3SPACE}: "+K$(ASC(MID$(D$,56
      ,1))-65): GOSUB 39000: GOSUB 33640             <EVR>
33530 PU$="CASS.NR.{4SPACE}: "+MID$(D$,57,4): GOS
      UB 39000: GOSUB 33640                          <VNV>
33540 PU$="LAUFZEIT{4SPACE}: "+MID$(D$,61,3)+" MI
      NUTEN": GOSUB 39000: GOSUB 33640               <8HP>
33550 PU$="FARBE{7SPACE}: ": IF MID$(D$,64,1)="1"
       THEN PU$=PU$+"JA"                             <40B>
33554 IF MID$(D$,64,1)="2" THEN PU$=PU$+"NEIN"       <BT2>
33556 GOSUB 39000: GOSUB 33640                       <1CB>
33560 PU$="DOLBY{7SPACE}: ": IF MID$(D$,65,1)="1"
       THEN PU$=PU$+"JA"                             <99T>
33564 IF MID$(D$,65,1)="2" THEN PU$=PU$+"NEIN"       <GM2>
33566 GOSUB 39000: GOSUB 33640                       <PFB>
33570 PU$="LONGPLAY{4SPACE}: ": IF MID$(D$,66,1)=
      "1" THEN PU$=PU$+"JA"                          <J8O>
33574 IF MID$(D$,66,1)="2" THEN PU$=PU$+"NEIN"       <OF2>
33576 GOSUB 39000: GOSUB 33640                       <HEB>
33580 PU$="STEREO{6SPACE}: ": IF MID$(D$,67,1)="1
      " THEN PU$=PU$+"JA"                            <6VT>
33584 IF MID$(D$,67,1)="2" THEN PU$=PU$+"NEIN"       <082>
33586 GOSUB 39000: GOSUB 33640                       <91B>
33590 PU$="HIFI{8SPACE}: ": IF MID$(D$,68,1)="1"
      THEN PU$=PU$+"JA"                              <BQR>
33594 IF MID$(D$,68,1)="2" THEN PU$=PU$+"NEIN"       <BH2>
33596 GOSUB 39000: GOSUB 33640                       <10B>
33600 PU$="BEMERKUNGEN : "+RIGHT$(D$,42): GOSUB 3
      9000: GOSUB 33640                              <NEP>
33610 PRINT#4: PRINT#4: PRINT#4: P%=P%+1             <GI6>
33620 IF P%=3 THEN FOR I2=1 TO 12: PRINT#4: NEXT
      : P%=0                                         <4H4>
33630 CLOSE 4: F8%=0: RETURN                         <2PD>
33640 PRINT#4,PU$: RETURN                            <LRH>
33650 :                                              <D9N>
33698 REM DRUCKERFEHLERBEHANDLUNG                    <70T>
33699 :                                              <916>
33700 CLOSE 4                                        <LGL>
33710 CHAR 1,0,24,"{6SPACE}DRUCKERFEHLER! BITTE F
      EHLER BESEITIGEN UND FUNKTION ERNEUT WFHLEN
       !{5SPACE}"+CHR$(7)                            <P7S>
33720 IF F8%=0 THEN DCLOSE #2 ON U(UD%)              <V1P>
33730 TRAP : CLOSE 4: SLEEP 3: GOTO 1100             <0Q8>
33740 :                                              <CPH>
34498 REM DATENSATZ SPEICHERN                        <105>
34499 :                                              <50M>
34500 FOR I=1 TO 4                                   <Q63>
34510 IF GG$(I)="" THEN BEGIN                        <FAH>
34520 CHAR 1,0,24,"{8SPACE}KEIN EINTRAG IM "+CHR$
      (34)+"MUV"+CHR$(34)+"-FELD VORHANDEN. BITTE
       DATEN KORRIGIEREN!{9SPACE}"+CHR$(7)           <G8F>
34530 SLEEP 3: F4%=I                                 <KJ2>
34540 CHAR 1,0,24,+TE$(1)+"{RVSON}"+TE$(2)+"{RVOF
      F}"+TE$(3)+TE$(4)+TE$(5): U%=2                 <JI9>
34550 GOTO 5130                                      <544>
34560 BEND                                           <FPE>
34570 NEXT                                           <08G>
34575 CHAR 1,0,24,Z6$                                <U34>
34580 CHAR 1,0,24,"{20SPACE}DATENSATZ NUMMER ": P
      RINT D;" WIRD GESPEICHERT!{19SPACE}";          <98R>
34590 D$=LEFT$(GG$(1)+Z6$,55)+GG$(2)+RIGHT$("0000
      "+GG$(3),4)+RIGHT$("000"+GG$(4),3)             <KNL>
34600 FOR I=5 TO 9                                   <SIL>
34610 IF LEFT$(GG$(I),1)="J" THEN D$=D$+"1": GOTO
       34640                                         <TDB>
34620 IF LEFT$(GG$(I),1)="N" THEN D$=D$+"2": GOTO
       34640                                         <ITB>
34630 D$=D$+"0"                                      <F0B>
34640 NEXT                                           <KA0>
34650 D$=D$+LEFT$(GG$(10)+Z6$,42)                    <15V>
34660 RECORD #2,D,1                                  <95T>
34670 IF DS<>0 THEN GOSUB 45020: GOTO 1100           <FHK>
34680 PRINT#2,CHR$(34)+D$                            <AEO>
34690 IF DS<>0 THEN GOSUB 45020: GOTO 1100           <FDN>
34700 RE%(D)=1                                       <HMM>
34702 IF D>N% THEN N%=D                              <GK9>
34703 IF F6%=2 THEN I=1: DO UNTIL VAL(RIGHT$(G$(I
      ),4))=D: I=I+1: LOOP : GOTO 34710              <QKL>
34705 I=1: DO UNTIL G$(I)="" OR I=N%: I=I+1: LOOP
                                                     <ANS>
34710 G$=RIGHT$("0000"+RIGHT$(STR$(D),LEN(STR$(D)
      )-1),4)                                        <41N>
34720 G$(I)=LEFT$(D$,15)+MID$(D$,56,5)+G$            <FHB>
34730 IF VAL(MID$(G$(D),17,4))>CA% THEN CA%=VAL(M
      ID$(G$(D),17,4))                               <2VF>
34735 IF U%=3 THEN GOSUB 33500                       <36E>
34740 IF F6%=2 THEN 6005                             <P1O>
34750 GOTO 5010                                      <444>
34997 :                                              <BHC>
34998 REM PROGRAMMDISKETTE EINLEGEN                  <7GV>
34999 :                                              <812>
35000 IF DM%<>DD% OR UM%<>UD% THEN 35050             <CD4>
35010 CHAR 1,0,22,Z5$: PRINT Z6$                     <DSO>
35020 PRINT "{GREEN,DOWN,9SPACE}BITTE DIE PROGRAM
      MDISKETTE EINLEGEN UND EINE TASTE DRXCKEN !
      {10SPACE}"+CHR$(7);                            <LK6>
35030 IF JOY(2)>127 THEN 35050                       <A09>
35040 GET TT$: IF TT$="" THEN 35030                  <1UV>
35050 RETURN                                         <0E7>
35097 :                                              <0G0>
35098 REM DATENDISKETTE EINLEGEN                     <209>
35099 :                                              <30E>
35100 IF DM%<>DD% OR UM%<>UD% THEN 35150             <ST6>
35110 CHAR 1,0,22,Z5$: PRINT Z6$                     <DKO>
35120 PRINT "{GREEN,DOWN,10SPACE}BITTE DIE DATEND
      ISKETTE EINLEGEN UND EINE TASTE DRXCKEN !{1
      2SPACE}"+CHR$(7);                              <GJI>
35130 IF JOY(2)>127 THEN 35150                       <QIB>
35140 GET TT$: IF TT$="" THEN 35130                  <1UT>
35150 RETURN                                         <0A7>
35160 :                                              <7OT>
```

```
35998 REM VERTIKALE MENUESTEUERUNG                         <6GR>
35999 :                                                    <A1A>
36000 U%=1                                                 <AHR>
36020 PRINT TAB(28),TE$(9)                                 <OMS>
36030 CHAR 1,0,BY: C$(U%)="{RVSON}"                        <9SN>
36040 FOR I1=1 TO TE: PRINT TAB(25)C$(I1)TE$(I1):
       PRINT : NEXT                                        <7NH>
36050 SI=1: BI=FL%: GOSUB 11000                            <NB2>
36110 IF JOY(2)<>0 THEN 36130                              <79D>
36120 GET T$: IF T$="" THEN 36110                          <JMC>
36130 T=PEEK(212): C$(U%)=""                               <SDJ>
36140 IF T=86 OR JOY(2)>127 THEN RETURN                    <IU6>
36150 IF T=84 OR JOY(2)=4 OR JOY(2)=5 OR JOY(2)=6
       THEN U%=1-(U%<TE)*U%                                <7B9>
36160 IF T=83 OR JOY(2)=1 OR JOY(2)=2 OR JOY(2)=8
       THEN U%=U%-1-TE*(U%=1)                              <EM3>
36170 IF T=64 THEN GOTO 23000                              <21R>
36180 GOTO 36030                                           <0HP>
36190 :                                                    <68R>
36498 REM HORIZONTALE MENUESTEUERUNG                       <30D>
36499 :                                                    <106>
36500 U%=1                                                 <2KR>
36510 CHAR 1,0,24: C$(U%)="{RVSON}"                        <44L>
36520 FOR I1=1 TO TE: PRINT C$(I1)TE$(I1);: NEXT
      : FOR I3=1 TO 500: NEXT                              <QRL>
36530 SI=1: BI=FL%: GOSUB 11000                            <NJ2>
36540 IF JOY(2)<>0 THEN 36560                              <G7G>
36550 GET T$: IF T$="" THEN 36540                          <001>
36560 T=PEEK(212): C$(U%)=""                          :
      REM NR. DER GEDRUECKTEN TASTE                        <FDR>
36570 IF T=83 OR JOY(2)>127 THEN RETURN
        : REM NR. CRSR HOCH                                <AMG>
36580 IF T=86 OR JOY(2)=2 OR JOY(2)=3 OR JOY(2)=4
       THEN U%=1-(U%<TE)*U%       : REM NR. CRSR RE
      CHTS                                                 <TER>
36590 IF T=85 OR JOY(2)=6 OR JOY(2)=7 OR JOY(2)=8
       THEN U%=U%-1-TE*(U%=1)    : REM NR. CRSR LI
      NKS                                                  <I59>
36600 GOTO 36510                                           <1HH>
36610 :                                                    <58N>
37997 REM QUICKSORT ALGORITHMUS                            <5CK>
37998 REM SORT.FELD: G$(), UNT.GR.: UF%, OB.GR.:
      OF%, STACK: S%(), STACKP.: S1%                       <3KV>
37999 :                                                    <E1Q>
38000 S1%=1: S%(1,1)=UF%: S%(1,2)=OF%                      <4VR>
38010 DO WHILE S1%<>0                                      <5NN>
38020 L%=S%(S1%,1): R%=S%(S1%,2): S1%=S1%-1                <M6A>
38030 DO WHILE R%>L%                                       <T54>
38040 P%=L%: Q%=R%: ZZ$=G$(INT(L%+R%)/2)                   <253>
38050 DO WHILE P%<=Q%                                      <1B4>
38060 IF G$(P%)<ZZ$ THEN P%=P%+1: GOTO 38060               <SPJ>
38070 IF G$(Q%)>ZZ$ THEN Q%=Q%-1: GOTO 38070               <VDG>
38080 IF P%<=Q% THEN XX$=G$(P%): G$(P%)=G$(Q%): G
      $(Q%)=XX$: P%=P%+1: Q%=Q%-1                          <C5V>
38090 LOOP                                                 <IQP>
38100 IF R%-P%>Q%-L% THEN BEGIN                            <H09>
38110 IF L%<Q% THEN S1%=S1%+1: S%(S1%,1)=L%: S%(S
      1%,2)=Q%                                             <AH1>
38120 L%=P%                                                <Q2C>
38130 BEND : ELSE BEGIN                                    <OCV>
38140 IF P%<R% THEN S1%=S1%+1: S%(S1%,1)=P%: S%(S
      1%,2)=R%                                             <6H0>
38150 R%=Q%                                                <UAT>
38160 BEND                                                 <F5E>
38170 LOOP : LOOP                                          <Q1R>
38180 RETURN                                               <J07>
38997 :                                                    <JIC>
38998 REM UMWANDLUNGSROUTINE CBM-ASCII AUF DRUCKE
      R-ASCII                                              <BHF>
38999 :                                                    <G22>
39000 IF CW%=0 THEN RETURN                                 <5MM>
39010 P$=""                                 : REM HILFS
      STRING                                               <9NF>
39020 FOR I1=1 TO LEN(PU$)                     : REM G$
      (A) = UNZUWANDELNDER STRING                          <JKE>
39030 P$=P$+PR$(ASC(MID$(PU$,I1,1)))  : REM UMWAN
      DLUNGSROUTINE                                        <45D>
39040 NEXT                                  : REM G$ =
      UMGEWANDELTER STRING                                 <E4T>
39050 PU$=P$                                               <2Q0>
39060 RETURN                                : REM WEITE
      R IM PROGRAMM                                        <1SJ>
39997 :                                                    <HI4>
39998 REM INPUTROUTINE                                     <AHB>
39999 :                                                    <I2A>
40000 ZL=0                                    : REM LT
      = ANZAHL ZEICHEN                                     <4DN>
40005 G$=""                                   : REM LT
      = ANZAHL ZEICHEN                                     <C3V>
40010 W=LEN(G$)                               : REM G$
      = UEBERGABESTRING                                    <5J4>
40020 IF JOY(2)<>0 THEN 40025                              <BKS>
40023 PRINT CHR$(27)"E";: GET GE$: IF GE$="" THEN
       40020                                               <E2C>
40025 T=PEEK(212): IF JOY(2)<>0 OR T>82 AND T<87
      THEN ZL(F4%)=W: GG$(F4%)=G$: GOTO 41000              <BLS>
40027 IF F3%=1 THEN IF T=7 AND GE$=CHR$(17) THEN
      ZL(F4%)=W: GG$(F4%)=G$: F4%=F4%+1: GOTO 401
      50                                                   <RQN>
40028 IF F3%=1 THEN IF T=7 AND GE$=CHR$(145) THEN
       ZL(F4%)=W: GG$(F4%)=G$: F4%=F4%-1: GOTO 40
      150                                                  <KL7>
40030 IF GE$=CHR$(13) AND W<>0 THEN 40140                  <E8P>
40040 IF ZL<W THEN IF T=2 AND GE$=CHR$(29) THEN Z
      L=ZL+1: PRINT GE$;: GOTO 40020                       <IUS>
40050 IF ZL>0 THEN IF T=2 AND GE$=CHR$(157) THEN
      ZL=ZL-1: PRINT GE$;: GOTO 40020                      <A65>
40055 IF F4%=2 AND ZL>0 THEN IF GE$=CHR$(20) THEN
       POKE 54784,10: POKE 54785,8: WINDOW 0,0,79
      ,24: CHAR 1,20,7,LEFT$(Z6$,15): WINDOW CX,C
      Y,CX+LT+1,CY: PRINT TAB(ZL)                          <L9R>
40060 IF ZL>0 THEN IF GE$=CHR$(20) THEN G$=LEFT$(
      G$,ZL-1)+RIGHT$(G$,W-ZL): ZL=ZL-1: GOTO 401
      30                                                   <1UN>
40070 IF ZL<W THEN IF W<LT THEN IF GE$=CHR$(148)
      THEN G$=LEFT$(G$,ZL)+CHR$(32)+RIGHT$(G$,W-Z
      L): GOTO 40130                                       <V0U>
40080 IF INSTR(Z$,GE$)=0 THEN 40010                        <5TB>
40100 IF ZL=LT THEN 40010                                  <4IU>
40110 ZL=ZL+1: IF ZL=W+1 THEN W=ZL                         <ALC>
40120 G$=LEFT$(G$,ZL-1)+GE$+RIGHT$(G$,W-ZL)                <B6F>
40125 IF F3%=1 AND F4%=2 THEN POKE 54784,10: POKE
       54785,8: WINDOW 0,0,79,24: CHAR 1,20,7,K$(
      ASC(G$)-65): WINDOW CX,CY,CX+LT+1,CY: PRINT
       TAB(ZL-1)                                           <L3T>
40130 PRINT GE$;: POKE 244,0: POKE 245,0: GOTO 40
      010                                                  <PNT>
40140 IF F3%=1 THEN ZL(F4%)=ZL: GG$(F4%)=G$: F4%=
      F4%+1                                                <EL9>
40150 POKE 54784,10: POKE 54785,8: RETURN   : REM
       CURSOR AUSBLENDEN                                   <H80>
40160 :                                                    <HQ5>
40998 REM HORIZONTALE MENUESTEUERUNG EINGABEMASKE          <DHN>
40999 :                                                    <S3I>
41000 IF F3%=0 THEN 40020                                  <IUP>
41005 IF F3%>1 THEN 41200                                  <J2P>
41010 WINDOW 0,0,79,24: POKE 54784,10: POKE 54785
      ,8: CHAR 1,0,24                                      <VPQ>
41020 IF T=83 OR JOY(2)>127 THEN 41100                     <MQV>
41030 IF T=86 OR JOY(2)=2 OR JOY(2)=3 OR JOY(2)=4
       THEN U%=1-(U%<TE)*U%                                <PN3>
41040 IF T=85 OR JOY(2)=6 OR JOY(2)=7 OR JOY(2)=8
       THEN U%=U%-1-TE*(U%=1)                              <B6Q>
41045 C$(U%)="{RVSON}": FOR I=1 TO TE: PRINT C$(I
      )TE$(I);: NEXT : FOR I3=1 TO 500: NEXT : C$
      (U%)=""                                              <H30>
41050 WINDOW CX,CY,CX+LT+1,CY: PRINT TAB(ZL): GOT
      O 40020                                              <BCL>
41100 IF F6%=3 THEN 8530                                   <HBP>
41105 IF F6%=2 THEN 6865                                   <B9T>
41110 IF F6%=1 THEN 6100                                   <TTA>
41120 IF U%=1 THEN BEGIN                                   <F9T>
41122 IF P%=0 THEN 41128                                   <MJH>
41124 CLOSE 4: IF SA%=0 THEN OPEN 4,5: ELSE OPEN
      4,4,SA%                                              <491>
41126 FOR I3=1 TO 72-P%*20: PRINT#4: NEXT : P%=0:
       CLOSE 4                                             <KKV>
41128 F3%=0: F6%=0: DCLOSE #2 ON U(UD%): TRAP : G
      OTO 1100                                             <MKU>
41129 BEND                                                 <TB6>
41130 IF F6%=0 AND U%=2 OR U%=3 THEN 34500                 <6EF>
41140 IF F6%=0 AND U%=4 THEN 32000                         <7C0>
41150 IF F6%=0 AND U%=5 THEN 31000                         <40C>
41160 :                                                    <JQD>
41198 REM ZURUECK IN'S MENUE                               <8H3>
41199 :                                                    <M2Q>
41200 IF T=83 OR JOY(2)>127 THEN POKE 54784,10: P
      OKE 54785,8: WINDOW 0,0,79,24                        <U95>
41210 IF F3%=2 THEN IF T=83 OR JOY(2)>127 THEN 10
      00                                                   <TL2>
41220 IF F3%=3 THEN IF T=83 OR JOY(2)>127 THEN 13
      00                                                   <C64>
41230 IF F3%=4 THEN IF T=83 OR JOY(2)>127 THEN 98
      00                                                   <HII>
41240 IF F3%=5 THEN IF T=83 OR JOY(2)>127 THEN 80
      00                                                   <4L6>
41250 IF F3%=6 THEN IF T=83 OR JOY(2)>127 THEN 70
      10                                                   <RTR>
41260 GOTO 40020                                           <UOO>
41270 :                                                    <GA3>
43998 REM FEHLERBEHANDLUNG TEILPROGRAMME LADEN             <EHR>
43999 :                                                    <Q3A>
44000 DOPEN #1,(FI$): IF DS=0 THEN DCLOSE #1: RET
      URN                                                  <OSA>
44010 PRINT "{CLR}L A D E F E H L E R !"+CHR$(7)           <3UC>
44020 PRINT "{DOWN}BITTE XBERPRXFEN SIE IHRE PROG
      RAMMDISKETTE"                                        <K1F>
44030 PRINT "{DOWN}UND 'BOOTEN' SIE DAS PROGRAMM
      ERNEUT !{5DOWN}"                                     <3M6>
44040 DCLOSE #1                                            <1L4>
44050 NEW                                                  <4SQ>
44060 :                                                    <OR1>
44998 REM FEHLERBEHANDLUNG                                 <9H7>
44999 :                                                    <K2I>
45000 DOPEN #1,(FI$),D(DM%),U(UM%): IF DS=0 THEN
      45050: ELSE GOTO 45020                               <BKK>
45010 DOPEN #1,(FI$),D(DD%),U(UD%): IF DS=0 THEN
      45050                                                <03R>
45020 CHAR 1,0,24,"{2SPACE}DISKETTENFEHLER : {BRO
      WN}"+DS$+"{GREEN} - BITTE FEHLER KORRIGIERE
      N!{3SPACE}"+CHR$(7)                                  <G3R>
45030 IF JOY(2)>127 THEN 45050                             <5PH>
45040 GET T$: IF T$="" THEN 45030                          <TNO>
45050 DR%=DS: DCLOSE #1 ON U(UD%): DCLOSE #2 ON U
      (UD%): : RETURN                                      <QT8>
45060 :                                                    <QR9>
45070 END                                                  <DPU>
```

Listing 1. (Schluß)

```
Name : video-characters  1000 1fff
-----------------------------------
1000 : 7e 66 66 66 66 66 7e 00  ab
1008 : 00 ff 80 ef 6c 6c 6c 6f  e1
1010 : 60 6f 6c 6c ef 80 ff 00  d3
1018 : fe 82 ba a2 ba 82 fe 00  16
1020 : 00 ff c0 ef 6c 6c 6c 6c  03
1028 : 6c 6c 6c 6c 6c 6c 6c 6c  28
1030 : 6c 6c 6c 6c ef c0 ff 00  80
1038 : 00 ff 00 ff 00 00 00 00  38
1040 : 00 00 00 00 ff 00 ff 00  40
1048 : 00 e0 30 98 cc 6c 6c 6c  92
1050 : 6c 6c 6c cc 98 30 e0 00  36
1058 : 6c 6c 6c 6f 60 6f 6c 6c  0f
1060 : 00 e0 30 98 cc 6c 6c 7c  ca
1068 : 7c 6c 6c cc 98 30 e0 00  5e
1070 : 00 fe c6 ee 6c 6c 6c 6c  33
1078 : 00 0f 18 33 66 6c 6c 6c  c1
1080 : 6c 6c 6c 66 33 18 0f 00  3a
1088 : 00 00 00 fc 0c fc 00 00  d1
1090 : 6c 6c 6c 6c ef c0 ff 00  e0
1098 : 00 f0 18 cc 3c 00 00 c0  f5
10a0 : c0 c0 00 3c cc 18 f0 00  99
10a8 : 00 00 00 3c cc 18 f0 00  81
10b0 : 00 1f 30 67 cc d8 d8 d8  e1
10b8 : d8 d8 d8 cc 67 30 1f 00  40
10c0 : 00 f0 18 cc 7c 00 00 00  a0
10c8 : 0f 1b 33 60 33 1b 0f 00  86
10d0 : 00 00 fe 06 fe 00 00 00  41
10d8 : 00 00 7f 60 7f 00 00 00  bc
10e0 : f0 d8 cc 06 cc d8 f0 00  88
10e8 : 18 30 3e 06 06 06 07 00  15
10f0 : 30 18 f8 c0 c0 c0 c0 00  98
10f8 : 00 00 00 00 00 00 00 ff  f8
1100 : 00 00 00 00 00 00 00 00  01
1108 : 00 07 06 06 06 3e 30 18  12
1110 : 00 c0 c0 c0 c0 f8 18 30  4d
1118 : 0c 06 03 01 00 00 00 00  08
1120 : 60 c0 80 00 00 00 00 00  01
1128 : 7f 60 60 60 60 60 7f 00  02
1130 : fe 06 06 06 06 06 fe 00  00
1138 : 08 08 08 00 00 00 00 00  47
1140 : 7f 7f 7f 7f 7f 7f 7f 00  41
1148 : fe fe fe fe fe fe fe 00  49
1150 : 08 2a 1c 3e 1c 2a 08 00  6f
1158 : 00 08 08 3e 08 08 00 00  e7
1160 : 00 00 00 00 00 08 08 10  e1
1168 : 00 00 00 7e 00 00 00 00  38
1170 : 00 00 00 00 00 18 18 00  92
1178 : 00 02 04 08 10 20 40 00  7f
1180 : 3c 42 46 5a 62 42 3c 00  e3
1188 : 08 18 28 08 08 08 3e 00  61
1190 : 3c 42 02 0c 30 40 7e 00  ee
1198 : 3c 42 02 1c 02 42 3c 00  1d
11a0 : 04 0c 14 24 7e 04 04 00  4c
11a8 : 7e 40 78 04 02 44 38 00  08
11b0 : 1c 20 40 7c 42 42 3c 00  a3
11b8 : 7e 42 04 08 10 10 10 00  1b
11c0 : 3c 42 42 3c 42 42 3c 00  5d
11c8 : 3c 42 42 3e 02 04 38 00  9f
11d0 : 00 00 08 00 00 08 00 00  13
11d8 : 00 00 08 00 00 08 08 10  5b
11e0 : 0e 18 30 60 30 18 0e 00  0e
11e8 : 00 00 7e 00 7e 00 00 00  70
11f0 : 70 18 0c 06 0c 18 70 00  74
11f8 : 3c 42 02 0c 10 00 10 00  99
1200 : 10 08 04 00 00 00 00 00  16
1208 : ff 7f 3f 1f 0f 07 03 01  b2
1210 : ff ff 00 00 00 00 00 00  0f
1218 : 00 00 00 00 ff ff ff ff  18
1220 : 00 00 00 00 0f 0f 0f 0f  e4
1228 : 0f 0f 0f 0f 00 00 00 00  65
1230 : 08 08 08 04 03 00 00 00  ef
1238 : 08 08 08 10 e0 00 00 00  57
1240 : 07 07 07 07 07 07 07 07  40
1248 : c0 c0 c0 c0 c0 c0 c0 c0  47
1250 : 03 03 03 03 03 03 03 03  50
1258 : 80 40 20 10 08 04 02 01  ad
1260 : 80 80 80 80 80 80 80 ff  5e
1268 : 01 02 04 08 10 20 40 80  71
1270 : ff 00 00 00 00 00 00 00  70
1278 : ff 80 80 80 80 80 80 80  f7
1280 : ff 01 01 01 01 01 01 01  7e
1288 : ff fe fc f8 f0 e0 c0 80  7f
1290 : 00 00 00 00 03 04 08 08  11
1298 : 00 00 00 00 f0 f0 f0 f0  d5
12a0 : 00 00 00 00 e0 10 08 08  5f
12a8 : e0 e0 e0 e0 e0 e0 e0 e0  a7
12b0 : ff ff ff 00 00 00 00 00  af
12b8 : f0 f0 f0 f0 00 00 00 00  7b
12c0 : 00 00 00 00 00 ff ff ff  c0
12c8 : 00 00 00 00 00 00 ff ff  c8
12d0 : f0 f0 f0 f0 f0 f0 f0 f0  cf
12d8 : 00 00 00 00 aa 55 aa 55  83
12e0 : 01 01 01 01 01 01 01 01  de
12e8 : 00 00 00 00 01 03 06 0c  41
12f0 : 00 00 00 00 00 80 c0 60  b8
12f8 : 00 00 00 00 00 00 00 ff  f8
1300 : 00 00 00 00 00 00 00 00  01
1308 : 08 08 08 08 08 08 08 08  08
1310 : 08 08 08 08 0f 00 00 00  10
1318 : 08 08 08 08 ff 00 00 00  27
1320 : 08 08 08 08 f8 00 00 00  bf
1328 : 08 08 08 08 0f 08 08 08  99
1330 : 00 00 00 00 ff 00 00 00  30
1338 : 08 08 08 08 f8 08 08 08  47
1340 : 00 00 00 00 0f 08 08 08  a2
1348 : 00 00 00 00 ff 08 08 08  b9
1350 : 00 00 00 00 f8 08 08 08  50
1358 : 08 08 08 08 ff 08 08 08  d8
1360 : 08 10 3c 42 7e 40 3c 00  a3
1368 : 18 24 20 70 20 21 5e 00  2d
1370 : 10 08 3c 42 7e 40 3c 00  b7
1378 : 08 10 20 00 00 00 00 00  91
1380 : 1c 22 4a 56 4c 20 1e 00  49
1388 : 00 00 42 42 42 46 ba 80  a3
1390 : 20 10 38 04 3c 44 3a 00  16
1398 : 10 08 42 42 42 46 3a 00  c4
13a0 : 10 28 38 04 3c 44 3a 00  22
13a8 : 08 14 3c 42 7e 40 3c 00  ed
13b0 : 08 14 08 08 08 08 1c 00  f7
13b8 : 08 14 3c 42 42 42 3c 00  49
13c0 : 08 14 42 42 42 46 3a 00  ea
13c8 : 1f 10 10 10 d0 30 10 00  c4
13d0 : 7f 21 10 08 10 21 7f 00  ed
13d8 : 5a 24 42 7e 42 42 42 00  e4
13e0 : 5a 24 42 42 42 24 18 00  cb
13e8 : 18 42 42 42 42 42 3c 00  21
13f0 : 3c 42 42 5c 42 42 5c 40  92
13f8 : 08 14 00 00 00 00 00 00  0b
1400 : 81 99 99 99 99 99 81 ff  54
1408 : ff 00 7f 10 93 93 93 90  2e
1410 : 9f 90 93 93 10 7f 00 ff  4c
1418 : 01 7d 45 5d 45 7d 01 ff  19
1420 : ff 00 3f 10 93 93 93 93  3c
1428 : 93 93 93 93 93 93 93 93  27
1430 : 93 93 93 93 10 3f 00 ff  df
1438 : ff 00 ff 00 ff ff ff ff  37
1440 : ff ff ff ff 00 ff 00 ff  3f
1448 : ff 1f cf 67 33 93 93 93  fd
1450 : 93 93 93 33 67 cf 1f ff  6a
1458 : 93 93 93 90 9f 90 93 93  a0
1460 : ff 1f cf 67 33 93 93 83  f5
1468 : 83 93 93 33 67 cf 1f ff  72
1470 : ff 01 39 11 93 93 93 93  ac
1478 : ff f0 e7 cc 99 93 93 93  2e
1480 : 93 93 93 99 cc e7 f0 ff  c5
1488 : ff ff ff 03 f3 03 ff ff  3e
1490 : 93 93 93 93 10 3f 00 ff  3f
1498 : ff 0f e7 33 c3 ff ff 3f  3a
14a0 : 3f 3f ff c3 33 e7 0f ff  a6
14a8 : ff ff ff c3 33 e7 0f ff  ce
14b0 : ff e0 cf 98 33 27 27 27  7e
14b8 : 27 27 27 33 98 cf e0 ff  2f
14c0 : ff 0f e7 33 83 ff ff ff  df
14c8 : f0 e4 cc 9f cc e4 f0 ff  09
14d0 : ff ff 01 f9 01 ff ff ff  5e
14d8 : ff ff 80 9f 80 ff ff ff  f3
14e0 : 0f 27 33 f9 33 27 0f ff  38
14e8 : e7 cf c1 f9 f9 f9 f8 ff  ba
14f0 : cf e7 07 3f 3f 3f 3f ff  48
14f8 : ff ff ff ff ff ff ff 00  f7
1500 : ff ff ff ff ff ff ff ff  ff
1508 : ff f8 f9 f9 f9 c1 cf e7  fe
1510 : ff 3f 3f 3f 3f 07 e7 cf  d2
1518 : f3 f9 fc fe ff ff ff ff  27
1520 : 9f 3f 7f ff ff ff ff ff  3f
1528 : 80 9f 9f 9f 9f 9f 80 ff  4d
1530 : 01 f9 f9 f9 f9 f9 01 ff  5f
1538 : f7 f7 f7 ff ff ff ff ff  29
1540 : 80 80 80 80 80 80 80 ff  3e
1548 : 01 01 01 01 01 01 01 ff  46
1550 : f7 d5 e3 c1 e3 d5 f7 ff  30
1558 : ff f7 f7 c1 f7 f7 ff ff  c8
1560 : ff ff ff ff ff f7 f7 ef  de
1568 : ff ff ff 81 ff ff ff ff  97
1570 : ff ff ff ff ff e7 e7 ff  4d
1578 : ff fd fb f7 ef df bf ff  71
1580 : c3 bd b9 a5 9d bd c3 ff  1c
1588 : f7 e7 d7 f7 f7 f7 c1 ff  ae
1590 : c3 bd fd f3 cf bf 81 ff  31
1598 : c3 bd fd e3 fd bd c3 ff  12
15a0 : fb f3 eb db 81 fb fb ff  f3
15a8 : 81 bf 87 fb fd bb c7 ff  47
15b0 : e3 df bf 83 bd bd c3 ff  bc
15b8 : 81 bd fb f7 ef ef ef ff  54
15c0 : c3 bd bd c3 bd bd c3 ff  22
15c8 : c3 bd bd c1 fd fb c7 ff  f0
15d0 : ff ff f7 ff ff f7 ff ff  8c
15d8 : ff ff f7 ff ff f7 f7 ef  54
15e0 : f1 e7 cf 9f cf e7 f1 ff  b1
15e8 : ff ff 81 ff 81 ff ff ff  5f
15f0 : 8f e7 f3 f9 f3 e7 8f ff  6c
15f8 : c3 bd fd f3 ef ff ef ff  56
1600 : ef f7 fb ff ff ff ff ff  ea
1608 : 00 80 c0 e0 f0 f8 fc fe  5d
1610 : 00 00 ff ff ff ff ff ff  10
1618 : ff ff ff ff 00 00 00 00  17
1620 : ff ff ff ff f0 f0 f0 f0  5b
1628 : f0 f0 f0 f0 ff ff ff ff  ea
1630 : f7 f7 f7 fb fc ff ff ff  70
1638 : f7 f7 f7 ef 1f ff ff ff  19
1640 : f8 f8 f8 f8 f8 f8 f8 f8  3f
1648 : 3f 3f 3f 3f 3f 3f 3f 3f  48
1650 : fc fc fc fc fc fc fc fc  4f
1658 : 7f bf df ef f7 fb fd fe  02
1660 : 7f 7f 7f 7f 7f 7f 7f 00  61
1668 : fe fd fb f7 ef df bf 7f  5f
1670 : 00 ff ff ff ff ff ff ff  70
1678 : 00 7f 7f 7f 7f 7f 7f 7f  f9
1680 : 00 fe fe fe fe fe fe fe  81
1688 : 00 01 03 07 0f 1f 3f 7f  91
1690 : ff ff ff ff fc fb f7 f7  0e
1698 : ff ff ff ff 0f 0f 0f 0f  5a
16a0 : ff ff ff ff 1f ef f7 f7  e0
16a8 : 1f 1f 1f 1f 1f 1f 1f 1f  a8
16b0 : 00 00 00 ff ff ff ff ff  b0
16b8 : 0f 0f 0f 0f ff ff ff ff  f5
16c0 : ff ff ff ff ff 00 00 00  bf
16c8 : ff ff ff ff ff ff 00 00  c7
16d0 : 0f 0f 0f 0f 0f 0f 0f 0f  d0
16d8 : ff ff ff ff 55 aa 55 aa  2c
16e0 : fe fe fe fe fe fe fe 00  e1
16e8 : ff ff ff ff fe fc f9 f3  8e
16f0 : ff ff ff ff ff 7f 3f 9f  27
16f8 : ff ff ff ff ff ff ff 00  f7
1700 : ff ff ff ff ff ff ff ff  ff
1708 : f7 f7 f7 f7 f7 f7 f7 f7  07
1710 : f7 f7 f7 f7 f0 ff ff ff  0f
1718 : f7 f7 f7 f7 00 ff ff ff  08
1720 : f7 f7 f7 f7 07 ff ff ff  80
1728 : f7 f7 f7 f7 f0 f7 f7 f7  b6
1730 : ff ff ff ff 00 ff ff ff  2f
1738 : f7 f7 f7 f7 07 f7 f7 f7  28
1740 : ff ff ff ff f0 f7 f7 f7  dd
1748 : ff ff ff ff 00 f7 f7 f7  d6
1750 : ff ff ff ff 07 f7 f7 f7  4f
1758 : f7 f7 f7 f7 00 f7 f7 f7  d7
1760 : f7 ef c3 bd 81 bf c3 ff  1c
1768 : e7 db df 8f df de a1 ff  a2
1770 : ef f7 c3 bd 81 bf c3 ff  28
1778 : f7 ef df ff ff ff ff ff  5f
1780 : e3 dd b5 a9 b3 df e1 ff  b6
1788 : ff ff bd bd bd b9 45 7f  6c
1790 : df ef c7 fb c3 bb c5 ff  09
1798 : ef f7 bd bd bd b9 c5 ff  6b
17a0 : ef d7 c7 fb c3 bb c5 ff  1d
17a8 : f7 eb c3 bd 81 bf c3 ff  62
17b0 : f7 eb f7 f7 f7 f7 e3 ff  68
17b8 : f7 eb c3 bd bd bd c3 ff  26
17c0 : f7 eb bd bd bd b9 c5 ff  95
17c8 : e0 ef ef ef 2f cf ef ff  cb
17d0 : 80 de ef f7 ef de 80 ff  b2
17d8 : a5 db bd 81 bd bd bd ff  cb
17e0 : a5 db bd bd bd db e7 ff  f4
17e8 : e7 bd bd bd bd bd c3 ff  ae
17f0 : c3 bd bd a3 bd bd a3 bf  4d
17f8 : f7 eb ff ff ff ff ff ff  e5
1800 : 3e 60 3c 66 3c 06 7c 00  30
1808 : 00 00 7e 66 7e 66 66 00  29
1810 : 00 00 7c 66 7c 66 7c 00  e9
1818 : 00 00 7e 66 60 66 7e 00  b8
1820 : 00 00 78 6c 66 6c 78 00  78
1828 : 00 00 7e 60 7c 60 7e 00  99
1830 : 00 00 7e 60 7c 60 60 00  28
1838 : 00 00 7c 60 6c 6c 7c 00  80
1840 : 00 00 66 66 7e 66 66 00  5b
1848 : 00 00 7e 18 18 18 7e 00  27
1850 : 00 00 7c 0c 0c 6c 3c 00  06
1858 : 00 00 66 6c 78 6c 66 00  04
1860 : 00 00 60 60 60 60 7e 00  87
1868 : 00 00 63 77 7f 6b 63 00  11
1870 : 00 00 66 76 7e 6e 66 00  ce
1878 : 00 00 7e 66 66 66 7e 00  78
1880 : 00 00 7e 66 7e 60 60 00  59
1888 : 00 00 7e 66 66 6e 7f 00  cd
1890 : 00 00 7e 66 7e 6c 66 00  e2
1898 : 00 00 7e 60 7e 06 7e 00  56
18a0 : 00 00 7e 18 18 18 18 00  e6
18a8 : 00 00 66 66 66 66 7e 00  a2
18b0 : 00 00 66 66 7e 3c 18 00  41
18b8 : 00 00 63 6b 7f 77 63 00  40
18c0 : 00 00 66 3c 18 3c 66 00  de
18c8 : 00 00 66 66 3c 18 18 00  14
18d0 : 00 00 7e 0c 18 30 7e 00  ee
18d8 : 3c 30 30 60 30 30 3c 00  ba
18e0 : 00 c0 60 30 18 0c 06 00  58
18e8 : 3c 0c 0c 06 0c 0c 3c 00  00
18f0 : 18 3c 7e 18 18 18 18 00  6c
18f8 : 00 00 00 00 00 00 00 ff  f8
1900 : 00 00 00 00 00 00 00 00  01
```

```
1908 : 18 18 18 18 18 00 18 00   17
1910 : 66 66 66 00 00 00 00 00   43
1918 : 66 66 ff 66 ff 66 66 00   4b
1920 : 08 3e 68 3e 0b 3e 08 00   ec
1928 : 00 66 6c 18 30 6c cc 00   13
1930 : 38 6c 6c 38 6f 6e 3b 00   18
1938 : 18 18 18 00 00 00 00 00   63
1940 : 3c 30 30 30 30 30 3c 00   1c
1948 : 3c 0c 0c 0c 0c 0c 3c 00   21
1950 : 18 db 3c ff 3c db 18 00   68
1958 : 00 18 18 7e 18 18 00 00   7d
1960 : 00 00 00 00 00 18 18 30   e2
1968 : 00 00 00 7e 00 00 00 00   38
1970 : 00 00 00 00 00 18 18 00   92
1978 : 00 06 0c 18 30 60 c0 00   8b
1980 : 7e 66 6e 7e 76 66 7e 00   31
1988 : 38 78 d8 18 18 18 7e 00   72
1990 : 7e 66 06 1c 30 60 7e 00   46
1998 : 7e 06 06 3e 06 06 7e 00   ed
19a0 : 60 60 6c 6c 7e 0c 0c 00   51
19a8 : 7e 60 78 0c 06 66 7e 00   84
19b0 : 7e 60 60 7e 66 66 7e 00   da
19b8 : 7e 06 0c 3c 18 18 18 00   67
19c0 : 7e 66 66 7e 66 66 7e 00   6e
19c8 : 7e 66 66 7e 06 06 7e 00   6d
19d0 : 00 00 18 00 00 18 00 00   97
19d8 : 00 00 18 00 00 18 18 30   60
19e0 : 0e 18 30 60 30 18 0e 00   0e
19e8 : 00 00 7e 00 7e 00 00 00   70
19f0 : 70 18 0c 06 0c 18 70 00   74
19f8 : 3c 66 06 1c 18 00 18 00   4e
1a00 : 30 18 0c 00 00 00 00 00   40
1a08 : 7e 66 66 7e 66 66 66 00   56
1a10 : 7c 66 66 7c 66 66 7c 00   74
1a18 : 7e 60 60 60 60 60 7e 00   ed
1a20 : 78 6c 66 66 66 6c 78 00   e0
1a28 : 7e 60 60 7c 60 60 7e 00   81
1a30 : 7e 60 60 7c 60 60 60 00   11
1a38 : 7e 66 60 6e 66 66 7e 00   63
1a40 : 66 66 66 7e 66 66 66 00   76
1a48 : 7e 18 18 18 18 18 7e 00   18
1a50 : 7e 0c 0c 0c 0c 6c 7c 00   6f
1a58 : 63 66 6c 78 6c 66 63 00   a0
1a60 : 60 60 60 60 60 60 7e 00   17
1a68 : 63 77 7f 6b 63 63 63 00   b3
1a70 : 66 76 76 7e 6e 6e 66 00   73
1a78 : fe c6 c6 c6 c6 c6 fe 00   02
1a80 : 7e 66 66 7e 60 60 60 00   25
1a88 : fe c6 c6 c6 d6 ce ff 00   58
1a90 : 7e 66 66 7e 78 6c 66 00   2f
1a98 : 7e 60 60 7e 06 06 7e 00   b9
1aa0 : 7e 18 18 18 18 18 18 00   d6
1aa8 : 66 66 66 66 66 66 7e 00   3b
1ab0 : 66 66 66 3c 3c 18 18 00   4f
1ab8 : 63 63 63 6b 7f 77 63 00   54
1ac0 : 66 66 3c 18 3c 66 66 00   fc
1ac8 : 66 66 66 3c 18 18 18 00   25
1ad0 : 7e 06 0c 18 30 60 7e 00   57
1ad8 : 66 00 7e 66 7e 66 66 00   5f
1ae0 : c6 00 fe c6 c6 c6 fe 00   dd
1ae8 : 66 00 66 66 66 66 7e 00   48
1af0 : 00 00 03 fe b6 36 36 00   87
1af8 : 00 00 00 00 00 00 00 ff   f8
1b00 : 00 00 00 00 00 00 00 00   01
1b08 : 18 18 18 18 18 18 18 18   08
1b10 : 18 18 18 18 1f 00 00 00   2f
1b18 : 18 18 18 18 ff 00 00 00   45
1b20 : 18 18 18 18 f8 00 00 00   dd
1b28 : 18 18 18 18 1f 18 18 18   99
1b30 : 00 00 00 00 ff 00 00 00   30
1b38 : 18 18 18 18 f8 18 18 18   46
1b40 : 00 00 00 00 1f 18 18 18   84
1b48 : 00 00 00 00 ff 18 18 18   9a
1b50 : 00 00 00 00 f8 18 18 18   31
1b58 : 18 18 18 18 ff 18 18 18   d7
1b60 : 0c 18 7e 60 7c 60 7e 00   e9
1b68 : 1c 36 30 78 30 33 de 00   d3
1b70 : 30 18 7e 60 7c 60 7e 00   1d
1b78 : 08 10 20 00 00 00 00 00   91
1b80 : 3c 66 de d6 dc 60 3e 00   4b
1b88 : 00 00 66 66 66 66 fe c0   06
1b90 : 30 18 7e 66 7e 66 66 00   ed
1b98 : 30 18 66 66 66 66 7e 00   ce
1ba0 : 1c 36 7e 66 7e 66 66 00   f8
1ba8 : 1c 36 7e 60 7c 60 7e 00   50
1bb0 : 1c 36 7e 18 18 18 7e 00   c6
1bb8 : 38 6c 7e 66 66 66 7e 00   26
1bc0 : 18 34 66 66 66 66 7e 00   ec
1bc8 : 1f 18 18 18 d8 78 18 00   ae
1bd0 : 7f 33 18 0c 18 33 7f 00   8a
1bd8 : 00 66 00 7e 66 7e 66 00   cf
1be0 : 00 66 00 7e 66 66 7e 00   77
1be8 : 00 66 00 66 66 66 7e 00   7c
1bf0 : 00 00 3c 66 6c 66 6c 60   39
1bf8 : 1c 36 00 00 00 00 00 00   30
1c00 : c1 9f c3 99 c3 f9 83 ff   cf
1c08 : ff ff 81 99 81 99 99 ff   e6
1c10 : ff ff 83 99 83 99 83 ff   36
1c18 : ff ff 81 99 9f 99 81 ff   77
1c20 : ff ff 87 93 99 93 87 ff   c7
1c28 : ff ff 81 9f 83 9f 81 ff   b6
1c30 : ff ff 81 9f 83 9f 9f ff   37
1c38 : ff ff 83 9f 93 93 83 ff   ef
1c40 : ff ff 99 99 81 99 99 ff   24
1c48 : ff ff 81 e7 e7 e7 81 ff   68
1c50 : ff ff 83 f3 f3 93 c3 ff   99
1c58 : ff ff 99 93 87 93 99 ff   ab
1c60 : ff ff 9f 9f 9f 9f 81 ff   38
1c68 : ff ff 9c 88 80 94 9c ff   be
1c70 : ff ff 99 89 81 91 99 ff   11
1c78 : ff ff 81 99 99 99 81 ff   77
1c80 : ff ff 81 99 81 9f 9f ff   a6
1c88 : ff ff 81 99 99 91 80 ff   42
1c90 : ff ff 81 99 81 93 99 ff   3d
1c98 : ff ff 81 9f 81 f9 81 ff   d9
1ca0 : ff ff 81 e7 e7 e7 e7 ff   59
1ca8 : ff ff 99 99 99 99 81 ff   ad
1cb0 : ff ff 99 99 81 c3 e7 ff   1e
1cb8 : ff ff 9c 94 80 88 9c ff   2f
1cc0 : ff ff 99 c3 e7 c3 99 ff   a1
1cc8 : ff ff 99 99 c3 e7 e7 ff   7b
1cd0 : ff ff 81 f3 e7 cf 81 ff   b1
1cd8 : c3 cf cf 9f cf cf c3 ff   f5
1ce0 : ff 3f 9f cf e7 f3 f9 ff   67
1ce8 : c3 f3 f3 f9 f3 f3 c3 ff   cf
1cf0 : e7 c3 81 e7 e7 e7 e7 ff   73
1cf8 : ff ff ff ff ff ff ff 00   f7
1d00 : ff ff ff ff ff ff ff ff   ff
1d08 : e7 e7 e7 e7 e7 ff e7 ff   f8
1d10 : 99 99 99 ff ff ff ff ff   dc
1d18 : 99 99 00 99 00 99 99 ff   e4
1d20 : f7 c1 97 c1 f4 c1 f7 ff   53
1d28 : ff 99 93 e7 cf 93 33 ff   3c
1d30 : c7 93 93 c7 90 91 c4 ff   47
1d38 : e7 e7 e7 ff ff ff ff ff   0d
1d40 : c3 cf cf cf cf cf c3 ff   63
1d48 : c3 f3 f3 f3 f3 f3 c3 ff   6e
1d50 : e7 24 c3 00 c3 24 e7 ff   37
1d58 : ff e7 e7 81 e7 e7 ff ff   33
1d60 : ff ff ff ff ff e7 e7 cf   dd
1d68 : ff ff ff 81 ff ff ff ff   97
1d70 : ff ff ff ff ff e7 e7 ff   4d
1d78 : ff f9 f3 e7 cf 9f 3f ff   65
1d80 : c3 99 91 81 89 99 c3 ff   19
1d88 : e7 e7 c7 e7 e7 e7 81 ff   15
1d90 : 81 99 f9 e3 cf 9f 81 ff   d9
1d98 : 81 f9 f9 c1 f9 f9 81 ff   42
1da0 : 9f 9f 93 93 81 f3 f3 ff   ee
1da8 : 81 9f 87 f3 f9 99 81 ff   cb
1db0 : 81 9f 9f 81 99 99 81 ff   85
1db8 : 81 f9 f3 c3 e7 e7 e7 ff   08
1dc0 : 81 99 99 81 99 99 81 ff   11
1dc8 : 81 99 99 81 f9 f9 81 ff   22
1dd0 : ff ff e7 ff ff e7 ff ff   08
1dd8 : ff ff e7 ff ff e7 e7 cf   4f
1de0 : f1 e7 cf 9f cf e7 f1 ff   b1
1de8 : ff ff 81 ff 81 ff ff ff   5f
1df0 : 8f e7 f3 f9 f3 e7 8f ff   6c
1df8 : c3 99 f9 e3 e7 ff e7 ff   a1
1e00 : cf e7 f3 ff ff ff ff ff   c0
1e08 : 81 99 99 81 99 99 99 ff   b9
1e10 : 83 99 99 83 99 99 83 ff   ab
1e18 : 81 9f 9f 9f 9f 9f 81 ff   42
1e20 : 87 93 99 99 99 93 87 ff   5f
1e28 : 81 9f 9f 83 9f 9f 81 ff   ce
1e30 : 81 9f 9f 83 9f 9f 9f ff   4f
1e38 : 81 99 9f 91 99 99 81 ff   0c
1e40 : 99 99 99 81 99 99 99 ff   09
1e48 : 81 e7 e7 e7 e7 e7 81 ff   77
1e50 : 81 f3 f3 f3 f3 93 83 ff   30
1e58 : 9c 99 93 87 93 99 9c ff   0f
1e60 : 9f 9f 9f 9f 9f 9f 81 ff   a8
1e68 : 9c 88 80 94 9c 9c 9c ff   1c
1e70 : 99 89 89 81 91 91 99 ff   6c
1e78 : 01 39 39 39 39 39 01 ff   ed
1e80 : 81 99 99 81 9f 9f 9f ff   da
1e88 : 01 39 39 39 29 31 00 ff   b7
1e90 : 81 99 99 81 87 93 99 ff   f0
1e98 : 81 9f 9f 81 f9 f9 81 ff   76
1ea0 : 81 e7 e7 e7 e7 e7 e7 ff   69
1ea8 : 99 99 99 99 99 99 81 ff   14
1eb0 : 99 99 99 c3 c3 e7 e7 ff   10
1eb8 : 9c 9c 9c 94 80 88 9c ff   1b
1ec0 : 99 99 c3 e7 c3 99 99 ff   83
1ec8 : 99 99 99 c3 e7 e7 e7 ff   6a
1ed0 : 81 f9 f3 e7 cf 9f 81 ff   48
1ed8 : 99 ff 81 99 81 99 99 ff   50
1ee0 : 39 ff 01 39 39 39 01 ff   e2
1ee8 : 99 ff 99 99 99 99 81 ff   87
1ef0 : ff ff fc 01 49 c9 c9 ff   58
1ef8 : ff ff ff ff ff ff ff 00   f7
1f00 : ff ff ff ff ff ff ff ff   ff
1f08 : e7 e7 e7 e7 e7 e7 e7 e7   07
1f10 : e7 e7 e7 e7 e0 ff ff ff   f0
1f18 : e7 e7 e7 e7 00 ff ff ff   ea
1f20 : e7 e7 e7 e7 07 ff ff ff   62
1f28 : e7 e7 e7 e7 e0 e7 e7 e7   b6
1f30 : ff ff ff ff 00 ff ff ff   2f
1f38 : e7 e7 e7 e7 07 e7 e7 e7   29
1f40 : ff ff ff ff e0 e7 e7 e7   fb
1f48 : ff ff ff ff 00 e7 e7 e7   f5
1f50 : ff ff ff ff 07 e7 e7 e7   6e
1f58 : e7 e7 e7 e7 00 e7 e7 e7   d8
1f60 : f3 e7 81 9f 83 9f 81 ff   d6
1f68 : e3 c9 cf 87 cf cc 21 ff   fc
1f70 : cf e7 81 9f 83 9f 81 ff   c2
1f78 : f7 ef df ff ff ff ff ff   5f
1f80 : c3 99 21 29 23 9f c1 ff   b4
1f88 : ff ff 99 99 99 99 01 3f   09
1f90 : cf e7 81 99 81 99 99 ff   32
1f98 : cf e7 99 99 99 99 81 ff   61
1fa0 : e3 c9 81 99 81 99 99 ff   47
1fa8 : e3 c9 81 9f 83 9f 81 ff   ff
1fb0 : e3 c9 81 e7 e7 e7 81 ff   99
1fb8 : c7 93 81 99 99 99 81 ff   49
1fc0 : e7 cb 99 99 99 99 81 ff   93
1fc8 : e0 e7 e7 e7 27 87 e7 ff   e1
1fd0 : 80 cc e7 f3 e7 cc 80 ff   15
1fd8 : ff 99 ff 81 99 81 99 ff   e0
1fe0 : ff 99 ff 81 99 99 81 ff   48
1fe8 : ff 99 ff 99 99 99 81 ff   53
1ff0 : ff ff c3 99 93 99 93 9f   a6
1ff8 : e3 c9 ff ff ff ff ff c2   45
```

Listing 2. Der neue Zeichensatz »VIDEO-CHARACTERS«. Bitte mit dem MSE (Seite 158) eingeben.

```
100 REM DELCO'S VIDEOARCHIV-SYSTEM                  <IIA>
102 :                                               <2SB>
103 REM STANDARD-KATEGORIEN FESTLEGEN               <IUB>
104 :                                               <3CD>
105 DIM K$(19)                                      <CKV>
106 K$(0)="AACTIONFILM{5SPACE}"                     <R1H>
107 K$(1)="BBERICHT/REPORT "                        <BLD>
108 K$(2)="HHISTORISCH{5SPACE}"                     <9AQ>
109 K$(3)="DDOKUMENTATION{2SPACE}"                  <NMC>
110 K$(4)="EEASTERN{8SPACE}"                        <FK0>
111 K$(5)="FFERNSEHFILM{4SPACE}"                    <E2K>
112 K$(6)="KKRIEGSFILM{5SPACE}"                     <VL8>
113 K$(7)="HHUMOR/KOMIK{4SPACE}"                    <NK5>
114 K$(8)="KKRIMINALFILM{3SPACE}"                   <BPO>
115 K$(9)="SSCIENCE FICTION"                        <DIK>
116 K$(10)="KKAMERAAUFNAHME "                       <BQK>
117 K$(11)="SSERIE{10SPACE}"                        <1TP>
118 K$(12)="MMUSIKVIDEO{5SPACE}"                    <8K6>
119 K$(13)="HHORRORFILM{5SPACE}"                    <VOM>
120 K$(14)="SSEXFILM{8SPACE}"                       <JHD>
121 K$(15)="PPORNOFILM{6SPACE}"                     <JN8>
122 K$(16)="WWESTERN{8SPACE}"                       <Q94>
123 K$(17)="TTRICKFILM{6SPACE}"                     <B73>
124 K$(18)="SSPIELFILM{6SPACE}"                     <FMU>
125 K$(19)="VVERSCHIEDENES{2SPACE}"                 <4LS>
126 :                                               <2SB>
127 DOPEN #1,"VIDEO-KATEGORIEN",W                   <BC0>
128 FOR A=0 TO 19: PRINT#1,K$(A): NEXT              <J94>
129 DCLOSE #1                                       <TP5>
130 :                                               <3SF>
131 END                                             <A5A>
```

Listing 3. »KATEGORIEN« zur Erstellung der sequentiellen Kategorien-Datei. Bitte mit dem Checksummer 128 (Seite 158) eingeben.

```
Name : video-centronics  1300 152a
-----------------------------------
1300 : 4c 05 13 05 01 a2 71 8e  3c
1308 : 1a 03 a2 13 8e 1b 03 a2  c2
1310 : a3 8e 1c 03 a2 13 8e 1d  99
1318 : 03 a2 bc 8e 1e 03 a2 13  18
1320 : 8e 1f 03 a2 d4 8e 20 03  9b
1328 : a2 13 8e 21 03 a2 ec 8e  32
1330 : 26 03 a2 13 8e 27 03 78  02
1338 : a9 ff 8d 03 dd ad 02 dd  b4
1340 : 09 04 8d 02 dd ad 00 dd  f6
1348 : 09 04 8d 00 dd a9 10 8d  3d
1350 : 0d dd ad 0d dd 58 60 48  0c
1358 : 8d 01 dd ad 00 dd 29 fb  1e
1360 : 8d 00 dd 09 04 8d 00 dd  ee
1368 : ad 0d dd 29 10 f0 f9 68  7a
1370 : 60 a6 b8 20 02 f2 d0 03  56
1378 : 4c 7f f6 a6 98 e0 0a 90  f0
1380 : 03 4c 7c f6 e6 98 a5 b8  e2
1388 : 9d 62 03 a5 b9 09 60 9d  6c
1390 : 76 03 a5 ba 9d 6c 03 cd  2d
1398 : 03 13 d0 02 18 60 c9 00  45
13a0 : 4c df ef 20 07 f2 f0 02  ac
13a8 : 18 60 20 12 f2 8a 48 a5  2b
13b0 : ba cd 03 13 f0 03 4c 94  f6
13b8 : f1 4c e4 f1 20 02 f2 f0  06
13c0 : 03 4c 82 f6 20 12 f2 a5  12
13c8 : ba cd 03 13 d0 03 4c 8b  f9
13d0 : f6 4c 10 f1 20 02 f2 f0  ee
13d8 : 03 4c 82 f6 20 12 f2 a5  2a
13e0 : ba cd 03 13 d0 03 4c 69  cd
13e8 : f1 4c 56 f1 8d 2a 15 48  e2
13f0 : a5 9a cd 03 13 f0 03 4c  13
13f8 : 7c ef 86 97 a5 b9 29 0f  eb
1400 : c9 10 d0 03 4c 2b 15 c9  6c
1408 : 00 f0 06 68 aa bd 2a 14  78
1410 : 48 68 20 57 13 ae 04 13  5c
1418 : f0 09 c9 0d d0 05 a9 0a  91
1420 : 20 57 13 a6 97 ad 2a 15  3f
1428 : 18 60 00 01 02 03 04 05  e3
1430 : 06 07 08 09 0a 0b 0c 0d  20
1438 : 0e 0f 10 11 12 13 14 15  28
1440 : 16 17 18 19 1a 1b 1c 1d  30
1448 : 1e 1f 20 21 22 23 24 25  38
1450 : 26 27 28 29 2a 2b 2c 2d  40
1458 : 2e 2f 30 31 32 33 34 35  48
1460 : 36 37 38 39 3a 3b 3c 3d  50
1468 : 3e 3f 40 61 62 63 64 65  20
1470 : 66 67 68 69 6a 6b 6c 6d  60
1478 : 6e 6f 70 71 72 73 74 75  68
1480 : 76 77 78 79 7a 5b 5c 5d  ae
1488 : 5e 5f 60 61 62 63 64 65  78
1490 : 66 67 68 69 6a 6b 6c 6d  80
1498 : 6e 6f 70 71 72 73 74 75  88
14a0 : 76 77 78 79 7a 7b 7c 7d  90
14a8 : 7e 7f 80 81 82 83 84 85  97
14b0 : 86 87 88 89 8a 8b 8c 8d  9f
14b8 : 8e 8f 90 91 92 93 94 95  a7
14c0 : 96 97 98 99 9a 9b 9c 9d  af
14c8 : 9e 9f a0 a1 a2 a3 a4 a5  b7
14d0 : a6 a7 a8 a9 aa ab ac ad  bf
14d8 : ae af b0 b1 b2 b3 b4 b5  c7
14e0 : b6 b7 b8 b9 ba 7b 7c 7d  4c
14e8 : 7e bf c0 41 42 43 44 45  78
14f0 : 46 47 48 49 4a 4b 4c 4d  e0
14f8 : 4e 4f 50 51 52 53 54 55  e8
1500 : 56 57 58 59 5a 5b 5c 5d  f0
1508 : de df e0 e1 e2 e3 e4 e5  f7
1510 : e6 e7 e8 e9 ea eb ec ed  ff
1518 : ee ef f0 f1 f2 f3 f4 f5  07
1520 : f6 f7 f8 f9 fa fb fc fd  0f
1528 : fe ff 9f 9f 9f 9f 80 ff  fb
```

Listing 4. »VIDEO-CENTRONICS« für Drucker am User-Port. Bitte mit dem MSE (Seite 158) eingeben.

```
100 REM DELCO'S VIDEOARCHIV SYSTEM                          <IIA>
102 :                                                       <2SB>
103 REM DRUCKERTABELLE FUER MPS 801 UND MPS 803             <IUB>
104 :                                                       <3CD>
105 DIM PR$(255)                                            <VQ1>
106 PR$(0)=CHR$(7)                                          <SU6>
107 PR$(1)=CHR$(14)                                         <808>
108 PR$(2)=CHR$(15)                                         <P4A>
109 PR$(3)=CHR$(0)                                          <5I0>
110 FOR A=4 TO 32: PR$(A)=CHR$(32): NEXT                    <UDR>
111 FOR A=33 TO 57: PR$(A)=CHR$(A): NEXT                    <94A>
112 FOR A=65 TO 90: PR$(A)=CHR$(A): NEXT                    <6NI>
113 FOR A=94 TO 185: PR$(A)=CHR$(A): NEXT                   <TLL>
114 FOR A=187 TO 191: PR$(A)=CHR$(A): NEXT                  <FVT>
115 FOR A=193 TO 255: PR$(A)=CHR$(A): NEXT                  <E3S>
116 PR$(60)=CHR$(58)                                        <8FJ>
117 PR$(61)=CHR$(61)                                        <B3S>
118 PR$(62)=CHR$(59)                                        <8VL>
119 PR$(63)=CHR$(63)                                        <BBP>
120 READ A: PR$(187)=CHR$(A): FOR A=1 TO 7: READ
    B: PR$(187)=PR$(187)+CHR$(B+128): NEXT                  <BLK>
121 READ A: PR$(187)=PR$(187)+CHR$(A)                       <E10>
122 READ A: PR$(219)=CHR$(A): FOR A=1 TO 7: READ
    B: PR$(219)=PR$(219)+CHR$(B+128): NEXT                  <G0K>
123 READ A: PR$(219)=PR$(219)+CHR$(A)                       <240>
124 READ A: PR$(188)=CHR$(A): FOR A=1 TO 7: READ
    B: PR$(188)=PR$(188)+CHR$(B+128): NEXT                  <V5A>
125 READ A: PR$(188)=PR$(188)+CHR$(A)                       <A1V>
126 READ A: PR$(220)=CHR$(A): FOR A=1 TO 7: READ
    B: PR$(220)=PR$(220)+CHR$(B+128): NEXT                  <7GA>
127 READ A: PR$(220)=PR$(220)+CHR$(A)                       <64V>
128 READ A: PR$(189)=CHR$(A): FOR A=1 TO 7: READ
    B: PR$(189)=PR$(189)+CHR$(B+128): NEXT                  <VL8>
129 READ A: PR$(189)=PR$(189)+CHR$(A)                       <A1U>
130 READ A: PR$(221)=CHR$(A): FOR A=1 TO 7: READ
    B: PR$(221)=PR$(221)+CHR$(B+128): NEXT                  <508>
131 READ A: PR$(221)=PR$(221)+CHR$(A)                       <6KS>
132 READ A: PR$(190)=CHR$(A): FOR A=1 TO 7: READ
    B: PR$(190)=PR$(190)+CHR$(B+128): NEXT                  <55U>
133 READ A: PR$(190)=PR$(190)+CHR$(A)                       <DPA>
134 READ A: PR$(64)=CHR$(A): FOR A=1 TO 7: READ B
    : PR$(64)=PR$(64)+CHR$(B+128): NEXT                     <C6R>
135 READ A: PR$(64)=PR$(64)+CHR$(A)                         <U62>
136 :                                                       <2C9>
137 DOPEN #2,"VIDEO-MPS 801{3SPACE}",W                      <6KG>
138 FOR A=32 TO 222: PRINT#2,CHR$(34)PR$(A): NEXT
                                                            <7KC>
139 DCLOSE #2                                               <TT4>
140 PRINT "{2DOWN}MPS 801 DATEN SIND GESPEICHERT!
    "                                                       <I9L>
141 :                                                       <040>
142 DOPEN #2,"VIDEO-MPS 803{3SPACE}",W                      <0MG>
143 FOR A=32 TO 222: PRINT#2,CHR$(34)PR$(A): NEXT
                                                            <70C>
144 DCLOSE #2                                               <TH4>
145 PRINT "{2DOWN}MPS 803 DATEN SIND GESPEICHERT!
    "                                                       <IDL>
146 :                                                       <1S7>
147 DATA 8,33,84,84,60,65,0,0,15 : REM KLEIN AE             <0RU>
148 DATA 8,125,10,10,10,125,0,0,15: REM GROSS AE            <N44>
149 DATA 8,57,68,68,68,57,0,0,15 : REM KLEIN OE             <0PC>
150 DATA 8,61,66,66,66,61,0,0,15 : REM GROSS OE             <B53>
151 DATA 8,61,64,64,61,64,0,0,15 : REM KLEIN UE             <U11>
152 DATA 8,61,64,64,64,61,0,0,15 : REM GROSS UE             <TL1>
153 DATA 8,126,1,41,41,22,0,0,15 : REM SZ                   <RGG>
154 DATA 8,10,85,85,85,40,0,0,15 : REM PARAGRAPH            <TM7>
```

Listing 5. »DRUCKERTAB-MPS« erzeugt die Druckertabellen für MPS 801 und 803 auf der Programmdiskette. Bitte mit dem Checksummer 128 (Seite 158) eingeben.

```
100 REM DELCO'S VIDEOARCHIV-SYSTEM                          <IIA>
102 :                                                       <2SB>
103 REM DRUCKERTABELLE FUER EPSON-DRUCKER                   <IUB>
104 :                                                       <3CD>
105 DIM PR$(255): E=27                                      <U5G>
106 FOR A=32 TO 64: PR$(A)=CHR$(A): NEXT                    <J93>
107 FOR A=65 TO 90: PR$(A)=CHR$(A+32): NEXT                 <KOT>
108 FOR A=94 TO 171: PR$(A)=CHR$(A): NEXT                   <8N1>
109 PR$(175)=CHR$(39)                                       <68D>
110 FOR A=187 TO 190: PR$(A)=CHR$(A-64): NEXT               <5KA>
111 PR$(191)=CHR$(94)                                       <EGU>
112 PR$(192)=CHR$(96)                                       <FBV>
113 FOR A=193 TO 221: PR$(A)=CHR$(A-128): NEXT              <SH4>
114 PR$(91)=CHR$(E)+CHR$(82)+CHR$(0)+CHR$(91)+CHR
    $(E)+CHR$(82)+CHR$(2)                                   <3D3>
115 PR$(92)=CHR$(E)+CHR$(82)+CHR$(0)+CHR$(92)+CHR
    $(E)+CHR$(82)+CHR$(2)                                   <4D1>
116 PR$(93)=CHR$(E)+CHR$(82)+CHR$(0)+CHR$(93)+CHR
    $(E)+CHR$(82)+CHR$(2)                                   <7D7>
117 PR$(172)=CHR$(E)+CHR$(82)+CHR$(1)+CHR$(123)+C
    HR$(E)+CHR$(82)+CHR$(2)                                 <RKQ>
118 PR$(173)=CHR$(E)+CHR$(82)+CHR$(3)+CHR$(35)+CH
    R$(E)+CHR$(82)+CHR$(2)                                  <DVA>
119 PR$(174)=CHR$(E)+CHR$(82)+CHR$(1)+CHR$(125)+C
    HR$(E)+CHR$(82)+CHR$(2)                                 <SNA>
120 PR$(175)=CHR$(39)                                       <500>
121 PR$(176)=CHR$(E)+CHR$(82)+CHR$(0)+CHR$(64)+CH
    R$(E)+CHR$(82)+CHR$(2)                                  <EEQ>
122 PR$(177)=CHR$(117)                                      <DKM>
123 PR$(178)=CHR$(E)+CHR$(82)+CHR$(1)+CHR$(64)+CH
    R$(E)+CHR$(82)+CHR$(2)                                  <HEM>
124 PR$(179)=CHR$(E)+CHR$(82)+CHR$(1)+CHR$(124)+C
    HR$(E)+CHR$(82)+CHR$(2)                                 <U1I>
125 PR$(180)=CHR$(97)+CHR$(8)+CHR$(94)                      <SC4>
126 PR$(181)=CHR$(101)+CHR$(8)+CHR$(94)                     <FGF>
127 PR$(182)=CHR$(105)+CHR$(8)+CHR$(94)                     <AG3>
128 PR$(183)=CHR$(111)+CHR$(8)+CHR$(94)                     <4HJ>
129 PR$(184)=CHR$(117)+CHR$(8)+CHR$(94)                     <8HR>
130 FOR A=1 TO 32: READ B: PR$(185)=PR$(185)+CHR$
    (B): NEXT                                               <KK0>
131 FOR A=1 TO 32: READ B: PR$(186)=PR$(186)+CHR$
    (B): NEXT                                               <CI0>
132 FOR A=1 TO 32: READ B: PR$(222)=PR$(222)+CHR$
    (B): NEXT                                               <TI3>
133 DATA 27,82,1,27,38,0,65,65,139,8,0,8,4,0,126,
    128,0,128,0,128,27,37,1,0,65,27,37,0,0,27,82,
    2 : REM WURZEL                                          <62H>
134 DATA 27,82,1,27,38,0,65,65,139,130,68,170,16,
    130,0,130,0,198,0,0,27,37,1,0,65,27,37,0,0,27
    ,82,2 : REM SUMMENZEICHEN                               <2BT>
135 DATA 27,82,1,27,38,0,65,65,139,3,60,2,0,2,0,2
    ,60,2,0,0,27,37,1,0,65,27,37,0,0,27,82,2 : RE
    M MY                                                    <NFA>
136 DOPEN #1,"VIDEO-EPSON{5SPACE}",W                        <440>
137 FOR A=32 TO 222: PRINT#1,CHR$(34)+PR$(A): NEX
    T                                                       <RI8>
138 DCLOSE #1                                               <TL5>
139 :                                                       <1K6>
140 END                                                     <54M>
```

Listing 6. »DRUCKERTAB-EPSON« erstellt die Druckertabelle für Epson-Drucker und Kompatible. Bitte mit dem Checksummer 128 (Seite 158) eingeben.

Der Computer wertet Versuche aus

Nach experimentellen Versuchen stehen Sie häufig vor dem Problem, die Ergebnisse mathematisch auszuwerten. Mit diesem Programm findet Ihr Computer zu experimentell gewonnenen Daten die zugehörige Gleichung.

Wenn Sie beispielsweise in einem Versuch Meßwerte für einen Fototransistor gewonnen haben, dann wollen Sie auch möglichst genau die Kennlinie für dieses Bauteil ermitteln. Normalerweise geht man so vor, daß man auf Millimeterpapier ein Koordinatenkreuz zeichnet, die Meßwerte einträgt und dann versucht, die Punkte durch eine »glatte« Kurve zu verbinden. Wenn Sie außerdem wissen möchten, ob es eine Gleichung gibt, die die experimentellen Ergebnisse hinreichend genau darstellt, kommen Sie nicht daran vorbei, umfangreiche Berechnungen anzustellen. Eine mühsame Arbeit, die der Computer für Sie erledigen kann. Mit dem Programm »Approxi 128« (Listing 1) – bitte beachten Sie dazu die Eingabehinweise auf Seite 159 – und einem C128 können Sie die Ergebnisse auf dem Bildschirm oder Plotter grafisch darstellen (Bild 1). Sie können auch beispielsweise die Buglinie eines Schiffsrumpfes auf diese Art entwerfen.

So läßt sich eine Gleichung finden

Wir wollen Ihnen auch erklären, wie sich aus den Meßergebnissen eine mathematische Gleichung finden läßt. Einige Kenntnisse der Mathematik sind dazu allerdings erforderlich.

Solange es sich bei der gesuchten Gleichung um eine stetige Funktion handelt, läßt sie sich in jedem Fall durch ein Polynom ausdrücken. Gleichungen, die man aufgrund experimenteller Daten ermittelt hat, nennt man »empirische Gleichungen«. Eine ideale empirische Gleichung muß die experimentellen Daten hinreichend genau darstellen und gleichzeitig einfach genug sein. Das heißt, sie darf nur so wenig Koeffizienten wie möglich enthalten. Diese beiden Bedingungen widersprechen sich jedoch. Man muß deshalb oft zwischen der Genauigkeit und der Einfachheit der Gleichung entscheiden. Da es keine Methode gibt, mit deren Hilfe man den Typ der rationellsten Gleichung für die Darstellung der gefundenen Daten angeben kann, behilft man sich dadurch, daß man die Meßergebnisse grafisch in Form von Punkten darstellt und diese durch eine »glatte« Kurve verbindet. Die Kurve soll dabei möglichst nahe an den gegebenen Punkten vorbeigeführt werden, und die Abweichungen sollen gleichmäßig positiv und negativ ausfallen. Aufgrund persönlicher Erfahrungen können Sie dann auf den Gleichungstyp schließen. Sobald der Gleichungstyp gewählt ist, sind die Koeffizienten zu bestimmen.

Die wahrscheinlichsten Werte der Koeffizienten findet man nach der »Methode der kleinsten Quadrate«. Hierbei geht man von folgendem aus:

Gegeben ist eine Anzahl von Datenpunkten. Gesucht wird ein sogenanntes »Approximationspolynom« des Grades m der Form

Nr.	Bestrahlungsstaerke E in (mW/qm)	Photonenstrom I in (mA)
1	.030	.016
2	.040	.032
3	.100	.063
4	.120	.111
5	.220	.216
6	.380	.390
7	.640	.804
8	1.090	1.570

```
Y=A+B*X+C*X↑2

A:-.020024
B: .908165
C: .454206
```

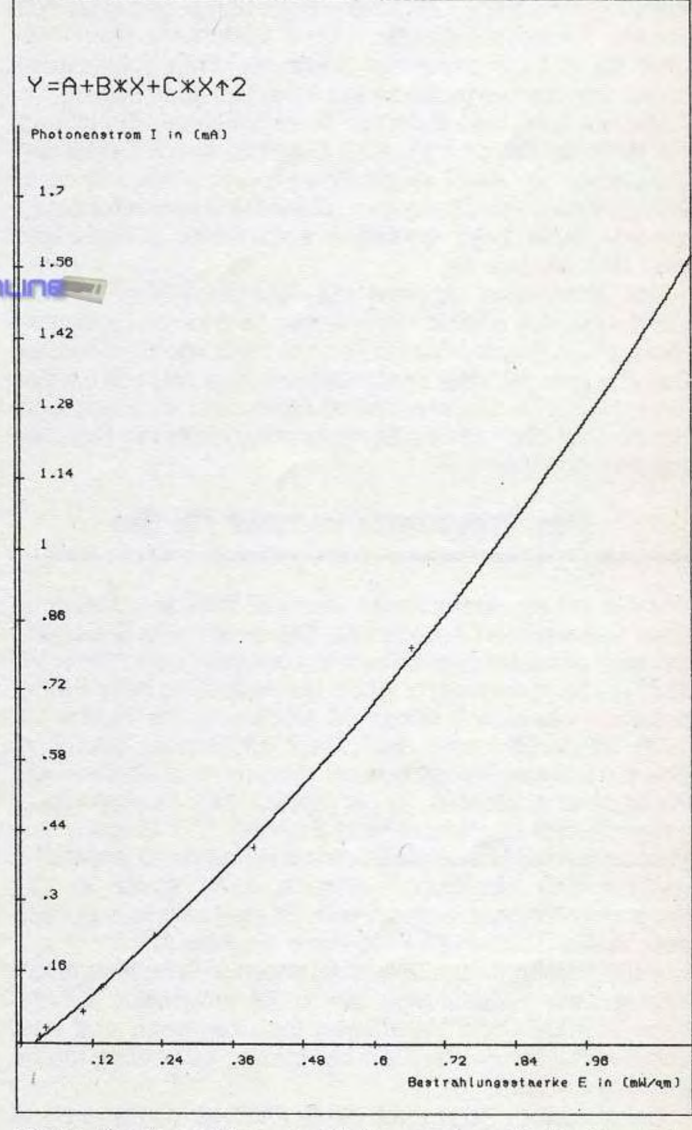


Bild 1. Die Auswertung der Meßergebnisse für die Kennlinie eines Fototransistors (mit dem Commodore-Plotter VC 1520)

$$P(x) = a_0 + a_1x + a_2x^2 + \ldots + a_mx^m$$

wobei der Fehler $E_i = y_i - P(x_i)$ für $i=1 \ldots n$ ist. Bei der Methode der kleinsten Quadrate wird die Summe der Feh-

ler so klein wie möglich gemacht. Daraus ergibt sich die Funktion:

$$f(a_0 \ldots a_m) = \sum_{i=1}^{n} (y_i - \sum_{j=0}^{m} a_j * \frac{j}{i})^2$$

Um das Minimum zu finden, bildet man die erste Ableitung für jedes a_j und setzt diese gleich 0.

$$\frac{df}{da_k} = -2 \sum_{i=1}^{n} ((y_i - \sum_{j=0}^{m} a_j * \frac{j}{i}) * \frac{k}{i}) = 0 \text{ für } k = 0,1,\ldots,m$$

Dieser Ausdruck kann umgeformt werden zu:

$$\sum_{i=1}^{n} y_i * \frac{k}{i} - \sum_{i=1}^{n} (\sum_{j=0}^{m} a_j * \frac{j}{i}) = 0$$

Dies ist ein System mit m+1 linearen Gleichungen und den Unbekannten a_0 , a_1 , ... , a_m.

Das Gleichungssystem wird in diesem Programm mit dem Gaußschen Algorithmus gelöst. Wie Sie sehen, sind umfangreiche Berechnungen notwendig, die Ihnen aber zum Glück der Computer abnimmt.

Wie schon erwähnt, soll die Gleichung möglichst einfach sein. Als einfachste funktionale Beziehung erweist sich die lineare Funktion y=a+bx. Diese Gleichung kann man unmittelbar dann anwenden, wenn die Meßergebnisse mit hinreichender Genauigkeit auf einer Geraden liegen.

Jedoch kann man auch von einer beliebigen Gleichung, die stets die Form g(x,y)=a+b f(x,y) hat, durch Einführung der neuen Variablen v=g(x,y); u=f(x,y) zu einer linearen Funktion v=a+b u übergehen. Dieses »Linearisierungsverfahren« findet beim Aufstellen empirischer Gleichungen sehr oft Verwendung.

Das Programm »Approxi 128« überprüft Polynome bis vierten Grades und fünf Funktionen, die durch Linearisierung auf ein Polynom ersten Grades zurückgeführt werden. Damit lassen sich fast alle Meßergebnisse grafisch darstellen, egal ob die Meßergebnisse nach exponentieller, logarithmischer oder nach einer gebrochen-linearen Funktion zusammenhängen.

Das Programm rechnet für Sie

Wie Sie sehen, nimmt Ihnen »Approxi 128« alle mathematisch aufwendigen Arbeiten ab. Die errechnete Gleichung läßt sich dann auf dem Bildschirm oder auf dem Plotter VC 1520 grafisch darstellen, wobei die Auflösung beim Plotter natürlich wesentlich höher ist. Übrigens, der Plotter VC 1520 ist bereits unter 200 Mark im Handel erhältlich. Ebenso können Sie mit diesem Programm auch beliebige Funktionen darstellen, sogar bis zu drei Funktionen in einem Koordinatenkreuz (Bild 2 und 4). Zur Eingabe von Funktionen muß der erste Punkt im Hauptmenü angewählt werden. Die Funktionen werden dann direkt in das Programm-Listing übernommen. Es sind bereits drei interessante Funktionen im Programm eingebaut.

Um die Berechnung zu beschleunigen wird im Programm zweimal der FAST-Modos des C128 eingesetzt. Einmal beim Einfügen von Funktionen ins Programm und zum anderen bei der Suche nach der idealen Gleichung für die Meßergebnisse.

Bei letzterem kann auch ohne FAST-Modus gerechnet werden, der Bildschirm bleibt dann eingeschaltet.

So gehen Sie mit dem Programm um

Das Programm besitzt drei Auswahlmenüs. Die einzelnen Funktionen werden entweder durch direktes Drücken einer Taste ausgeführt oder durch Ansteuern mit den Cursortasten und <RETURN>. Ebenso werden alle Eingaben mit <RETURN> abgeschlossen.

Die Berechnung der mathematischen Funktion zu den Meßdaten erfolgt unter dem Punkt »Polynomapproximation« des ersten Menüs. Dabei müssen Sie die Anzahl der Wertepaare vorgeben und diese eingeben.

Anschließend können Sie entweder selbst eine Funktionsart wählen oder den Computer die günstigste Funktionsart suchen lassen. Dabei wird die Summe der Abweichungen (im Quadrat) als Fehler angegeben. Die einzelnen Abweichungen werden quadriert, damit sich positive und negative Abweichungen nicht gegenseitig aufheben können.

Ausgabe auf Plotter

Während der Darstellung der Funktion auf dem Bildschirm ist der Bildschirmrahmen schwarz. Ist der Vorgang beendet, wird der Bildschirmrahmen blau. Anschließend wartet der Computer so lange, bis die RETURN-Taste gedrückt wird. Im Bild 3 sehen Sie ein Beispiel für eine Funktion, die nach willkürlich eingegebenen Wertepaaren gesucht wurde. Da das Programm im hochauflösenden Grafikmo-

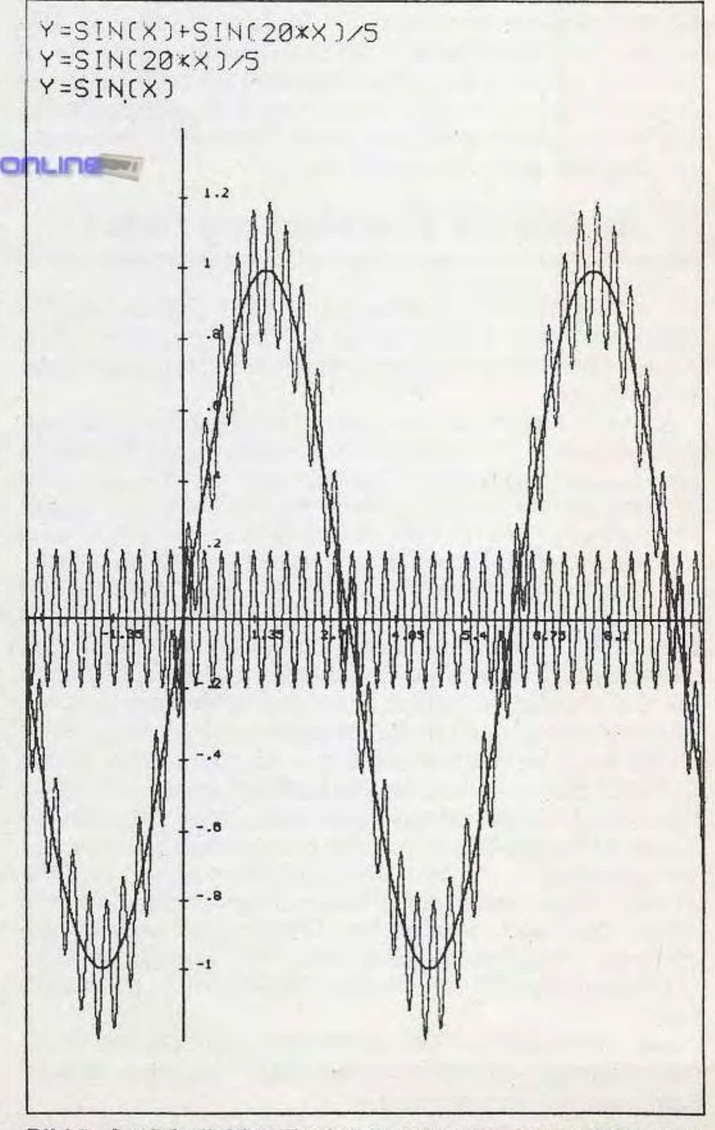


Bild 2. Auch beliebige Funktionen können eingegeben und grafisch dargestellt werden – bis zu drei Funktionen lassen sich in einem Koordinatenkreuz generieren

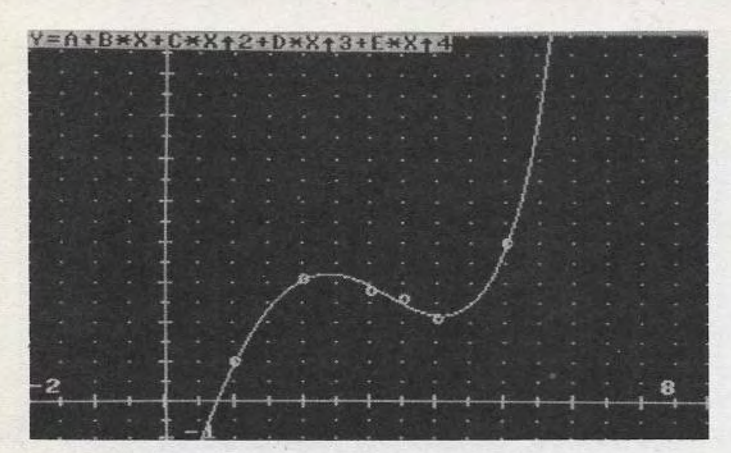


Bild 3. Die Bildschirmausgabe der Funktion, die der Computer nach der Eingabe von willkürlichen sechs Wertepaaren berechnet hat. Die Konstanten für die Funktion (links oben) werden nach <RETURN> ausgegeben.

dus arbeitet, kann es manchmal zu Farbüberläufen kommen (an den Schnittpunkten einer Kurve mit den Koordinatenachsen bzw. zweier Kurven miteinander) was aber nicht weiter störend ist.

Mit »Approxi 128« lassen sich beliebige Funktionen darstellen, auch Funktionen mit Definitionslücken. Der Befehl »TRAP« verhindert bei unzulässigen Werten einen Programmabbruch. Für die Darstellung der Funktion ist der Definitions- und Wertebereich frei wählbar unter der Voraussetzung, daß die Koordinatenachsen in diesem Bereich liegen. Der erste einzugebende Wert muß daher zu jeder Koordinatenachse negativ sein.

Bei der Darstellung auf dem Plotter wird die maximale horizontale Auflösung von 480 Punkten ausgenutzt, in vertikaler Richtung beträgt die Auflösung 600 Punkte. Der Abstand der Markierungen auf den beiden Achsen beträgt genau 10 mm. Dadurch können auch Zwischenwerte leicht aus der Plotter-Grafik ausgemessen werden.

Das Programmlisting ist mit REM-Anmerkungen gut kommentiert und damit leicht nachvollziehbar.

Insgesamt erhalten Sie mit »Approxi 128« ein Werkzeug, das Sie sowohl für wissenschaftliche Zwecke als auch für den Hobby-Bereich nutzen können.

(Joachim Stolte/kn)

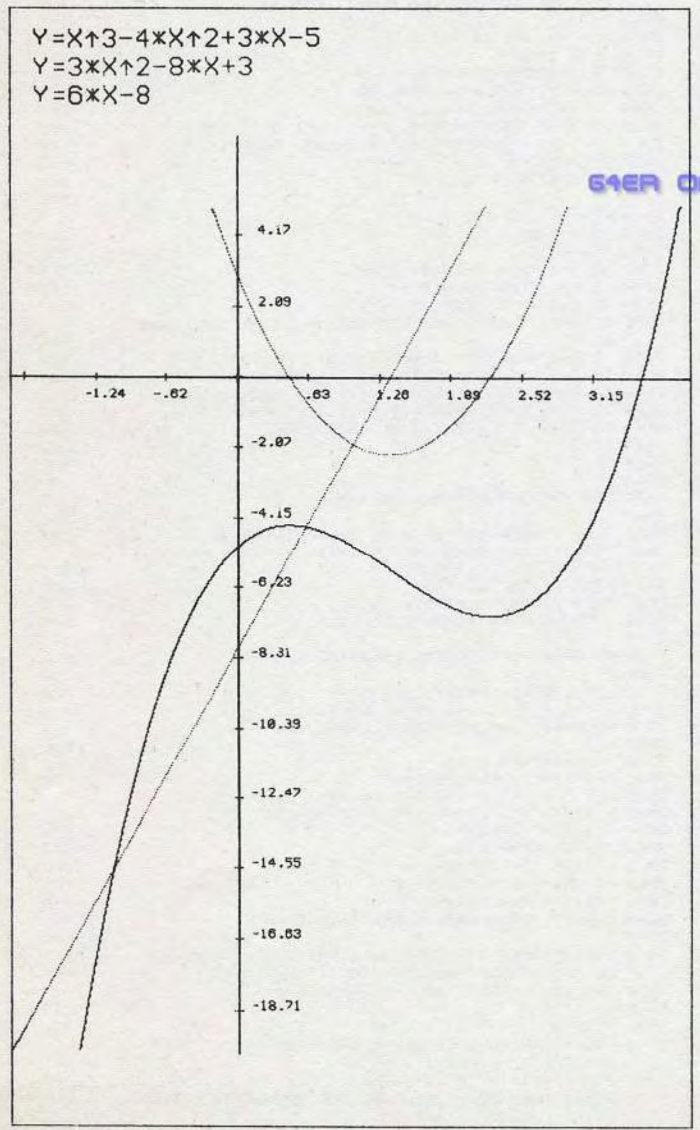
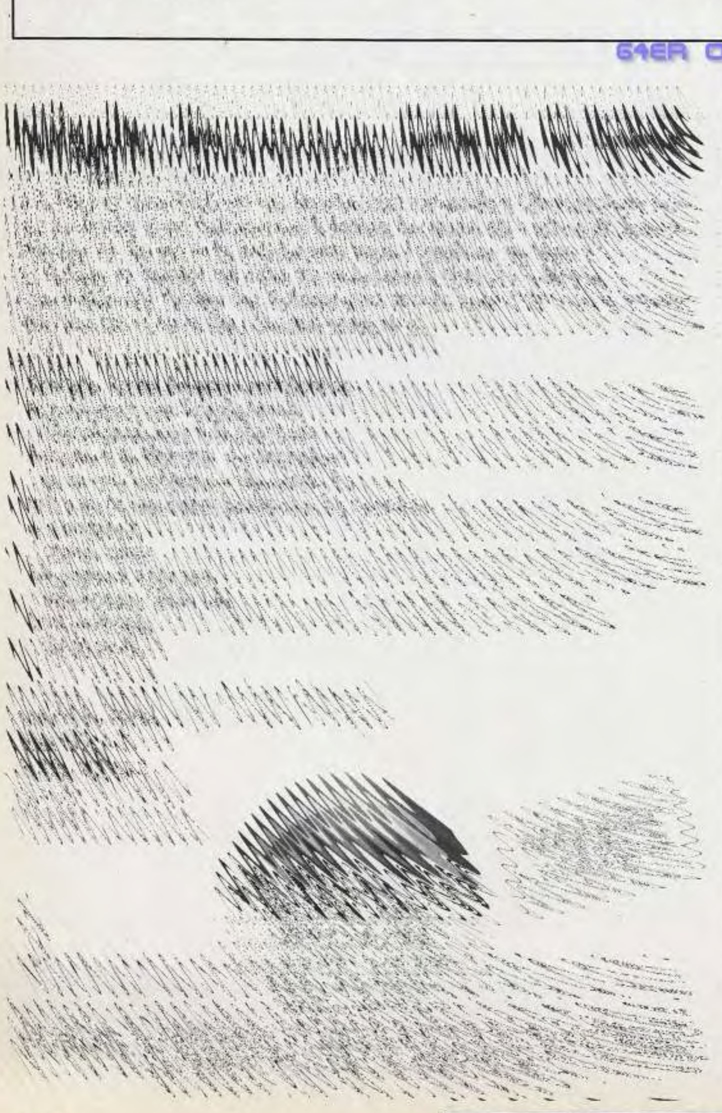


Bild 4. Fast alle experimentell ermittelten Kurven sind durch entsprechende Funktionen darstellbar. Hier drei Beispiele von elementaren Kurvenverläufen.

```
1000 REM FUNKTIONENPLOTTER & POLYNOMAPPROXIMATION      <34C>
1010 REM ********************* 1987 *************      <3CD>
1020 REM ****************************************      <3KE>
1030 COLOR 0,7,0: COLOR 4,7,0: COLOR 1,8,7: GRAPH
     IC 1,1: GRAPHIC 0                                  <SKJ>
1040 SLOW                                               <EDG>
1050 GOSUB 8010                                         <PUB>
1060 :                                                  <209>
1070 PRINT CHR$(147);CHR$(14);CHR$(18);"{7SPACE}A
     USWAHLMENUE{8SPACE}";CHR$(146)                     <F1N>
1080 PRINT : PRINT "(1) EINGABE NEUER FUNKTIONEN"      <CR1>
1090 PRINT : PRINT "(2) AUSGABE AUF BILDSCHIRM"        <N5D>
1100 PRINT : PRINT "(3) AUSGABE AUF PLOTTER"           <F05>
1110 PRINT : PRINT "(4) POLYNOMAPPROXIMATION"          <D0E>
1120 PRINT : PRINT "(5) PROGRAMM BEENDEN"              <GQQ>
1130 GET KEY A$: SP=VAL(A$)                             <5NM>
1140 IF A$<"1" OR A$>"5" THEN 1130                      <CGL>
1150 ON SP GOTO 1190,5680,1410,2750,6330                <Q49>
1160 :                                                  <30D>
1170 REM *** EINGABE DER FUNKTIONEN ***                 <2C9>
1180 :                                                  <40H>
1190 PRINT CHR$(147): PRINT "ANZAHL DER FUNKTIONE
     N ";: GOSUB 6340: ZZ=W: PRINT                      <HU3>
1200 IF ZZ<1 OR ZZ>3 THEN 1190                          <NNM>
1210 FOR I=1 TO 3: A$(I)="": NEXT                       <5I1>
1220 FOR I=1 TO ZZ                                      <563>
1230 PRINT : INPUT "Y=";A$(I)                           <KBR>
1240 NEXT I                                             <DL4>
1250 :                                                  <1G7>
1260 REM *** NEUE FUNKTIONEN INS PROGRAMM EINFUEG
     EN ***                                             <3KE>
1270 :                                                  <0G3>
1280 PRINT CHR$(147);"8010 DEF FNA(X)=";A$(1)           <R1D>
1290 PRINT "8020 DEF FNB(X)=";A$(2)                     <D59>
1300 PRINT "8030 DEF FNC(X)=";A$(3)                     <M51>
1310 PRINT "8040 A$(1)=";CHR$(34);"Y=";A$(1);CHR$
     (34)                                               <BSH>
1320 PRINT "8050 A$(2)=";CHR$(34);"Y=";A$(2);CHR$
     (34)                                               <2DH>
1330 PRINT "8060 A$(3)=";CHR$(34);"Y=";A$(3);CHR$
     (34)                                               <QUH>
1340 PRINT "8070 ZZ=";ZZ                                <9EV>
1350 PRINT "RUN"                                        <D3F>
1360 SYS 50861,19 : FOR I = 1 TO 8 : SYS 50861,13
      : NEXT                                            <E3I>
1370 FAST : END : REM NEUE ZEILEN UEBER TASTATURP
     UFFER-TRICK INS PROGRAMM NEHMEN                    <HJ0>
1380 :                                                  <60P>
1390 REM *** AUSGABE DER FUNKTIONEN AUF PLOTTER *
     **                                                 <0S3>
1400 :                                                  <50L>
1410 FOR X=1 TO 7: CLOSE X: NEXT                        <EHT>
1420 OPEN 1,6,1                                         <4P1>
1430 OPEN 2,6,2                                         <25D>
1440 OPEN 3,6,3                                         <3HG>
1450 OPEN 4,6,0                                         <DC9>
1460 OPEN 6,6,6: PRINT#6,1                              <FQU>
1470 :                                                  <6GR>
1480 GOSUB 6500 : REM *** EINGABE ***                   <87B>
1490 :                                                  <1G7>
1500 REM *** BERECHNUNG DER X,Y-ACHSEN ***              <1K6>
1510 :                                                  <4GJ>
1520 IF A=0 THEN A=-B*.007: XF=1                        <QAD>
1530 IF B=0 THEN B=-A*.007: XF=1                        <HU0>
1540 XS=(B-A)/480                                       <MG5>
1550 YS=(D-C)/600                                       <DT9>
```

Listing 1. Das Programm »Approxi 128« berechnet die mathematische Funktion zu Ihren Versuchsergebnissen. Bitte mit dem Checksummer 128 (Seite 158) eingeben

```
1560 Y0=INT(ABS(A/XS)+.5)                                  <JVC>
1570 MA=D/YS: MI=C/YS: X0=INT(ABS(D/YS)+.5)                <QNK>
1580 IF C>0 THEN MI=0                                      <6LH>
1590 PRINT#1,"M",0,-50: PRINT#3,1                          <1D4>
1600 :                                                     <70T>
1610 REM *** AUSGABE DER FUNKTIONSTERME ***                <0C1>
1620 :                                                     <60P>
1630 PRINT#6,0                                             <01L>
1640 FOR I=1 TO ZZ                                         <B63>
1650 PRINT#2,4-I: PRINT#4," ";A$(I)                        <7LP>
1660 NEXT                                                  <AD0>
1670 PRINT#6,1                                             <25L>
1680 :                                                     <30D>
1690 REM *** ZEICHNET X-ACHSE ***                          <2C9>
1700 :                                                     <60P>
1710 PRINT#4: PRINT#4: PRINT#3,0                           <2HI>
1720 PRINT#1,"M",0,-X0                                     <VUP>
1730 PRINT#4: PRINT#2,0                                    <0BD>
1740 PRINT#1,"D",479,0: PRINT#1,"D",0,0                    <7NJ>
1750 X1=Y0-INT(Y0/50)*50                                   <62R>
1760 :                                                     <50L>
1770 FOR X=0 TO 8                                          <PMI>
1780 X2=X1+X*50                                            <V8J>
1790 PRINT#1,"M",X2,-3                                     <901>
1800 PRINT#1,"D",X2,3                                      <FVK>
1810 NEXT X                                                <70C>
1820 :                                                     <B11>
1830 REM *** BESCHRIFTET X-ACHSE ***                       <7SV>
1840 :                                                     <B1D>
1850 X2=X1: Z=INT((A+X1*XS)*100+.5)/100                    <I18>
1860 :                                                     <A19>
1870 FOR X=0 TO 8                                          <MMI>
1880 IF XF=1 AND X=0 THEN 1940                             <C57>
1890 X2=X1+50*X-8                                          <7UH>
1900 IF X2<9 THEN 1940                                     <10Q>
1910 PRINT#1,"M",X2,-15                                    <06B>
1920 IF ABS(Z)<0.05 THEN PRINT#4,"{2SPACE}";: GOT
     O 1940                                                <0B8>
1930 PRINT#4,Z;                                            <1SL>
1940 Z=INT((Z+50*XS)*100+.5)/100                           <JDS>
1950 IF Z>1E3 THEN Z=INT(Z*10+.5)/10                       <7UA>
1960 NEXT X                                                <F5C>
1970 :                                                     <BHF>
1980 REM *** ZEICHNET Y-ACHSE ***                          <5KM>
1990 :                                                     <CHJ>
2000 Y1=INT(D/YS+.5)                                       <GNH>
```



```
2010 Y2=INT(Y1/50)*50                                      <ETS>
2020 PRINT#1,"M",Y0,Y1+50                                  <OAE>
2030 PRINT#1,"D",Y0,Y1-600: PRINT#1,"D",Y0,Y1+50           <4NH>
2040 :                                                     <50L>
2050 REM *** BESCHRIFTET Y-ACHSE ***                       <1C5>
2060 :                                                     <40H>
2070 Z=INT((Y2*YS)*100+.5)/100                             <RJQ>
2080 YS=(D-C)/600                                          <KP9>
2090 FOR X=0 TO 11                                         <R1L>
2100 Y3=Y2-50*X                                            <LG0>
2110 PRINT#1,"M",Y0-3,Y3                                   <NI8>
2120 PRINT#1,"D",Y0+3,Y3                                   <VR0>
2130 IF ABS(Z)<0.05 THEN 2160                              <C5V>
2140 IF Y0>440 THEN PRINT#1,"M",Y0-45,Y3                   <JQT>
2150 PRINT#4," ";Z;                                        <65U>
2160 Z=INT((Z-50*YS)*100+.5)/100                           <49G>
2170 NEXT X                                                <70C>
2180 :                                                     <209>
2190 IF PO=0 THEN 2400                                     <30D>
2200 :                                                     <50L>
2210 REM *** BEI POLYNOMAPPROX. BENENNUNG DER ACH
     SEN ***                                               <1C5>
2220 :                                                     <40H>
2230 PRINT#1,"M",Y0+10,Y1+40: PRINT#4,NY$;" IN ("
     ;EY$;")";                                             <493>
2240 N$=NX$+" IN ("+EX$+")"                                <LRT>
2250 H=470-LEN(N$)*6                                       <5U8>
2260 PRINT#1,"M",H,-30: PRINT#4,N$;                        <56I>
2270 PRINT#2,2                                             <29S>
2280 :                                                     <105>
2290 REM *** KENNZEICHNUNG DER MESSWERTE ***                <3CD>
2300 :                                                     <40H>
2310 FOR L=1 TO N                                          <75D>
2320 X=(XP(L)-A)/XS: Y=YP(L)/YS                            <ONO>
2330 IF X<1 OR X>481 THEN 2360                             <VOS>
2340 PRINT#1,"M",X-2,Y-4                                   <FH9>
2350 PRINT#4,"+";                                          <0DB>
2360 NEXT L                                                <DD6>
2370 :                                                     <7GV>
2380 REM *** AUSGABE DER FUNKTIONEN ***                    <3KE>
2390 :                                                     <0G3>
2400 FOR V=1 TO ZZ                                         <VP3>
2410 TRAP 2450                                             <JET>
2420 IF V=1 THEN Y=FN A(A)/YS                              <3B3>
2430 IF V=2 THEN Y=FN B(A)/YS                              <FCJ>
2440 IF V=3 THEN Y=FN C(A)/YS                              <BE3>
2450 IF Y>MA THEN PRINT#1,"M",0,MA: GOTO 2480              <H5E>
2460 IF Y<MI THEN PRINT#1,"M",0,MI: GOTO 2480              <L5M>
2470 PRINT#1,"M",0,Y : Z=0                                 <N25>
2480 PRINT#2,4-V                                           <DJQ>
2490 :                                                     <7GV>
2500 FOR I=1 TO 479                                        <11B>
2510 TRAP 2620                                             <J6U>
2520 X=A+I*XS                                              <T79>
2530 IF V=1 THEN Y=FN A(X)/YS                              <TCR>
2540 IF V=2 THEN Y=FN B(X)/YS                              <98B>
2550 IF V=3 THEN Y=FN C(X)/YS                              <L9R>
2560 IF QQ=1 THEN QQ=0: PRINT#1,"M",I,Y: GOTO 263
     0                                                     <E3L>
2570 IF Z=0 AND Y<MI THEN PRINT#1,"D",I,MI: Z=1            <5D7>
2580 IF Z=0 AND Y>MA THEN PRINT#1,"D",I,MA: Z=1            <98N>
2590 IF Y>MA THEN PRINT#1,"M",I,MA: GOTO 2630              <S2I>
2600 IF Y<MI THEN PRINT#1,"M",I,MI: GOTO 2630              <01A>
2610 PRINT#1,"D",I,Y : Z=0: GOTO 2630                      <G98>
2620 QQ=1                                                  <AL3>
2630 NEXT I,V                                              <N0L>
2640 :                                                     <30D>
2650 REM *** ZEICHNE RAHMEN ***                            <2C9>
2660 :                                                     <209>
2670 TRAP : PRINT#1,"M",0,MI-50: PRINT#2,0                 <C84>
2680 PRINT#1,"D",0,MA+140: PRINT#1,"D",479,MA+140
     : PRINT#1,"D",479,MI-50                               <0V9>
2690 PRINT#1,"D",0,MI-50                                   <3KV>
2700 PRINT#1,"M",0,MI-120                                  <BCT>
2710 IF PO=1 THEN 4200: ELSE 1000                          <PAM>
2720 :                                                     <105>
2730 REM *** POLYNOMAPPROXIMATION ***                      <3CD>
2740 :                                                     <209>
2750 CLR : PO=1: PRINT CHR$(147)                           <PNB>
2760 DIM R%(10),SX(8),XY(4),A(55)                          <419>
2770 REM *** FUNKTIONENBEZEICHUNG ***                      <2C9>
2780 :                                                     <40H>
2790 FK$(1)="Y=A+B*X"                                      <RH1>
2800 FK$(2)="Y=A+B*X+C*X↑2"                                <PTF>
2810 FK$(3)="Y=A+B*X+C*X↑2+D*X↑3"                          <QJS>
2820 FK$(4)="Y=A+B*X+C*X↑2+D*X↑3+E*X↑4"                    <E2I>
2830 FK$(5)="Y=A*EXP(B*X)                                  <Q4L>
2840 FK$(6)="Y=A*X↑B                                       <SAO>
2850 FK$(7)="Y=1/(A+B*X)                                   <ORA>
2860 FK$(8)="Y=X/(A+B*X)                                   <BJJ>
2870 FK$(9)="Y=A*LN(B*X)"                                  <JHM>
2880 FK$(10)="COMPUTER WAEHLT FUNKTION"                    <49I>
2890 :                                                     <BHF>
2900 PRINT CHR$(147);CHR$(18);CHR$(155);"{2SPACE}
     POLYNOMAPPROXIMATION{3SPACE}";CHR$(158)               <N3V>
2910 REM *** EINGABE DER WERTEPAARE ***                    <4SJ>
2920 :                                                     <F1T>
2930 PRINT                                                 <30U>
2940 PRINT "ANZAHL DER WERTEPAARE{2SPACE}";: GOSU
     B 6340: N=W: PRINT : PRINT                            <VIL>
2950 DIM X(N),Y(N),XP(N),YP(N)                             <TA7>
2960 PRINT CHR$(18);" EINGABE DER MESSWERTE ";CHR
     $(146): PRINT : PRINT                                 <SCH>
2970 FOR L=1 TO N                                          <E5D>
2980 PRINT L".{3SPACE}X= ";: GOSUB 6340: X(L)=W            <FN6>
```

```
2990 PRINT TAB(23);"Y= ";: GOSUB 6340: Y(L)=W: PR
     INT : PRINT                                        <R7Q>
3000 XP(L)=X(L): YP(L)=Y(L)                             <2SO>
3010 NEXT L                                             <5B6>
3020 :                                                  <40H>
3030 REM *** FUNKTIONENAUSWAHL ***                      <1S7>
3040 GOSUB 7020: A=10                                   <NN0>
3050 FOR L=1 TO N: X(L)=XP(L): Y(L)=YP(L): NEXT         <ECG>
3060 FOR L=1 TO 10: R%(L)=0: NEXT : SP=1: R%(1)=1
     8                                                  <F2U>
3070 PRINT CHR$(147)                                    <RR9>
3080 PRINT CHR$(19);CHR$(18);CHR$(155);"{2SPACE}P
     OLYNOMAPPROXIMATION{3SPACE}";CHR$(158)             <DHL>
3090 :                                                  <1G7>
3100 FOR L=1 TO 10                                      <MHT>
3110 L$=RIGHT$("{2SPACE}"+STR$(L),3)+". "               <S4N>
3120 PRINT                                              <BDU>
3130 PRINT CHR$(R%(L));L$;FK$(L)                        <R9U>
3140 NEXT L                                             <TF6>
3150 PRINT CHR$(155)                                    <RR9>
3160 PRINT CHR$(18);"DERZEIT ANGEWAEHLTER MENUEPU
     NKT ";SP;CHR$(157);" "                             <M00>
3170 GET KEY A$                                         <4OD>
3180 R%(SP)=0                                           <44F>
3190 IF ASC(A$)=13 THEN 3260                            <4S6>
3200 IF A$=CHR$(17) THEN SP=SP+1+A*(SP=A)               <63C>
3210 IF A$=CHR$(145) THEN SP=SP-1-A*(SP=1)              <3OP>
3220 IF A$=CHR$(19) THEN SP=1                           <UUS>
3230 R%(SP)=18                                          <9G2>
3240 GOTO 3080                                          <3OK>
3250 :                                                  <5GN>
3260 TRAP 6980 : REM *** BEI FEHLER VERZWEIGE ***       <5L8>
3270 :                                                  <4GJ>
3280 ON SP GOSUB 5010,5050,5090,5130,5150,5260,53
     70,5470,5570,6630                                  <CVE>
3290 :                                                  <3GF>
3300 REM +++ BERECHNUNG DER FUNKTION +++                <0K2>
3310 :                                                  <6GR>
3320 REM *** BERECHNUNG DER SUMMEN ***                  <040>
3330 :                                                  <7GV>
3340 FOR L=1 TO N: SY=SY+Y(L): NEXT                     <VPC>
3350 :                                                  <4GJ>
3360 FOR K=1 TO 2*M                                     <NS0>
3370 FOR L=1 TO N: SX(K)=SX(K)+X(L)↑K: NEXT             <2M9>
3380 NEXT K                                             <TR5>
3390 :                                                  <2GB>
3400 FOR K=1 TO M                                       <UQS>
3410 FOR L=1 TO N: XY(K)=XY(K)+Y(L)*X(L)↑K: NEXT        <MP1>
3420 NEXT K                                             <TV5>
3430 :                                                  <0G3>
3440 REM *** AUFSTELLUNG DER KOEFFIZIENTENMATRIX
     ***                                                <248>
3450 :                                                  <3GF>
3460 NG=M+1: B(1)=SY                                    <JAD>
3470 :                                                  <2GB>
3480 FOR L=2 TO NG: B(L)=XY(L-1): NEXT                  <QVT>
3490 A(11)=N                                            <R94>
3500 FOR L=2 TO NG: A(10+L)=SX(L-1): NEXT               <NBK>
3510 FOR Q=2 TO NG                                      <0A1>
3520 FOR L=1 TO NG: A(Q*10+L)=SX((Q-2)+L): NEXT         <S5L>
3530 NEXT Q                                             <6K8>
3540 :                                                  <209>
3550 REM *** LOESEN DES LINEAREN GLEICH.-SYST. **
     *                                                  <2SB>
3560 :                                                  <30D>
3570 FOR MG=1 TO NG                                     <PAV>
3580 P=A(MG*11): IF P=0 THEN 3700                       <HER>
3590 FOR L=MG TO NG                                     <BNF>
3600 A(MG*10+L)=A(MG*10+L)/P: HA(L)=A(MG*10+L)          <U9B>
3610 NEXT L                                             <546>
3620 B(MG)=B(MG)/P: HB(L)=B(MG): HB=B(MG)               <BMT>
3630 IF MG>=NG THEN 3690                                <APM>
3640 FOR Q=MG+1 TO NG                                   <LP9>
3650 V=-A(Q*10+MG)                                      <7N6>
3660 FOR L=MG TO NG: A(Q*10+L)=A(Q*10+L)+HA(L)*V:
      NEXT                                              <DTA>
3670 B(Q)=B(Q)+HB*V                                     <GNA>
3680 NEXT Q                                             <EP8>
3690 NEXT MG: GOTO 3800                                 <7E2>
3700 HL=MG+1                                            <Q91>
3710 IF HL>NG THEN 3780                                 <KN9>
3720 FOR L=MG TO NG                                     <LNF>
3730 HZ(HL*10+L)=A(MG*10+L): A(MG*10+L)=A(HL*10+L
     ): A(HL*10+L)=HZ(HL*10+L)                          <R2N>
3740 NEXT L                                             <T36>
3750 BH=B(MG): B(MG)=B(HL): B(HL)=BH                    <ROL>
3760 IF A(MG)*11=0 THEN HL=HL+1: GOTO 3710              <4K8>
3770 GOTO 3580                                          <30U>
3780 PRINT CHR$(147)"KOEFFIZIENTEN KOENNEN NICHT"
     : PRINT "BERECHNET WERDEN"                         <I6C>
3790 GOTO 1000                                          <20K>
3800 C(NG)=B(NG)                                        <TUB>
3810 FOR L=NG-1 TO 1 STEP -1                            <DAP>
3820 FOR Q=L+1 TO NG                                    <IUJ>
3830 B(L)=B(L)-C(Q)*A(L*10+Q)                           <386>
3840 NEXT Q                                             <E98>
3850 C(L)=B(L)                                          <UVP>
3860 NEXT L                                             <TV6>
3870 :                                                  <EHR>
3880 ON SP GOSUB 6460,6460,6460,6460,5220,5330,54
     40,5540,5640                                       <5F8>
3890 :                                                  <9H7>
3900 REM *** AUSGABE DER ERGEBNISSE ***                 <5KM>
3910 :                                                  <CHJ>
3920 IF FE=1 THEN 4100                                  <580>
3930 PRINT CHR$(147)                                    <QNC>
3940 PRINT CHR$(19);CHR$(18);CHR$(155);"{2SPACE}P
     OLYNOMAPPROXIMATION{3SPACE}";CHR$(158)             <CDL>
3950 PRINT : PRINT "ERGEBNIS FUER": PRINT : PRINT
      CHR$(18);FK$(SP)                                  <VQ1>
3960 PRINT : A$(1)=FK$(SP)                              <LSS>
3970 FOR L=1 TO NG                                      <GNP>
3980 IF ABS(C(L))<1E-07 THEN C(L)=0                     <1I2>
3990 C(L)=INT(C(L)*1E+07+.5)/1E+07                      <D7D>
4000 PRINT : PRINT CHR$(64+L);"=";C(L)                  <K35>
4010 NEXT L                                             <5K6>
4020 :                                                  <A19>
4030 REM *** BERECHNUNG DES FEHLERS ***                 <6SR>
4040 :                                                  <915>
4050 S=0                                                <RRB>
4060 FOR L=1 TO N                                       <BLD>
4070 S=S+(YP(L)-FN A(XP(L)))↑2: S=INT(S*1E6+.5)/1
     E6                                                 <VMO>
4080 NEXT                                               <I60>
4090 :                                                  <FHV>
4100 IF FG=1 THEN 6730                                  <5CS>
4110 TRAP                                               <99D>
4120 :                                                  <B1D>
4130 PRINT CHR$(155)                                    <QFF>
4140 PRINT : PRINT : PRINT "FEHLER = ";S                <CNU>
4150 PRINT : PRINT CHR$(18);"WEITER MIT >RETURN<"       <PIV>
4160 GET KEY A$: IF ASC(A$)<>13 THEN 4160               <A6T>
4170 :                                                  <9H7>
4180 REM *** MENUE ***                                  <4KI>
4190 :                                                  <EHR>
4200 PRINT CHR$(147)                                    <A7E>
4210 PRINT CHR$(19);CHR$(18);CHR$(155);"{2SPACE}P
     OLYNOMAPPROXIMATION{3SPACE}";CHR$(158)             <SJL>
4220 PRINT : PRINT "(1) AUSGABE AUF PLOTTER"            <VI5>
4230 PRINT : PRINT "(2) AUSGABE AUF BILDSCHIRM"         <KDD>
4240 PRINT : PRINT "(3) EINGABE NEUER MESSWERTE"        <Q2U>
4250 PRINT : PRINT "(4) NEUE FUNKTION WAEHLEN"          <QL3>
4260 PRINT : PRINT "(5) MESSWERTE UEBERPRUEFEN"         <N7U>
4270 PRINT : PRINT "(6) FUNKTION ZEIGEN"                <NKA>
4280 PRINT : PRINT "(7) ZURUECK INS HAUPT-MENUE"        <OC1>
4290 GET KEY A$: W=VAL(A$): IF W<1 OR W>7 THEN 42
     90                                                 <BVS>
4300 ON W GOTO 4340,5680,2730,3030,7080,3900,1000       <BGA>
4310 :                                                  <8H3>
4320 REM *** AUSGABE AUF PLOTTER ***                    <74S>
4330 :                                                  <9H7>
4340 OPEN 1,6,1: OPEN 2,6,2: OPEN 3,6,3: OPEN 4,6
     ,0: OPEN 6,6,6: PRINT#6,1                          <LO8>
4350 PRINT#3,2: PRINT#2,0                               <MD8>
4360 PRINT CHR$(147)                                    <AFF>
4370 PRINT "UEBERSCHRIFT (MAX. 20 ZEICHEN)";CHR$(
     17)                                                <L2E>
4380 INPUT A$                                           <ANF>
4390 IF LEN(A$)>20 THEN 4360                            <JCT>
4400 :                                                  <F1T>
4410 PRINT#4: PRINT#4,A$                                <E91>
4420 PRINT#3,0: PRINT#2,1                               <5B0>
4430 :                                                  <EHR>
4440 PRINT : INPUT "NAME DER GROESSE X";NX$             <C4G>
4450 PRINT : INPUT "EINHEIT VON X";EX$                  <2AV>
4460 PRINT : INPUT "NAME DER GROESSE Y";NY$             <CCH>
4470 PRINT : INPUT "EINHEIT VON Y";EY$                  <66E>
4480 :                                                  <B1D>
4490 PRINT#4,"{5SPACE}NR.";: PRINT#1,"M",100,0: P
     RINT#4,NX$;                                        <P14>
4500 PRINT#1,"M",300,0: PRINT#4,NY$                     <34Q>
4510 PRINT#1,"M",100,0: PRINT#4,"IN (";EX$;")";         <JMJ>
4520 PRINT#1,"M",300,0: PRINT#4,"IN (";EY$;")"          <CNN>
4530 PRINT#4: PRINT#4: PRINT#2,2                        <B9H>
4540 :                                                  <C1H>
4550 REM *** AUSGABE DER MESSWERTE AUF PLOTTER **
     *                                                  <5SN>
4560 :                                                  <D1L>
4570 FOR L=1 TO N                                       <E5D>
4580 N$="{6SPACE}"+STR$(L): N$=RIGHT$(N$,6)             <F4E>
4590 IF XP(L)<0 THEN Z=-1E-4: ELSE Z=1E-4               <5IM>
4600 X=INT(XP(L)*1E3+.5)/1E3+Z                          <H8D>
4610 IF YP(L)<0 THEN Z=-1E-4: ELSE Z=1E-4               <6AH>
4620 Y=INT(YP(L)*1E3+.5)/1E3+Z                          <GSF>
4630 X$="{9SPACE}"+STR$(X)                              <H0J>
4640 Y$="{9SPACE}"+STR$(Y)                              <562>
4650 X$=LEFT$(RIGHT$(X$,11),10): IF XP(L)=0 THEN
     X$="{5SPACE}0.000"                                 <CIL>
4660 Y$=LEFT$(RIGHT$(Y$,11),10): IF YP(L)=0 THEN
     Y$="{5SPACE}0.000"                                 <4ML>
4670 PRINT#4,N$;                                        <LC1>
4680 PRINT#1,"M",110,0: PRINT#4,X$;: PRINT#1,"M",
     310,0: PRINT#4,Y$                                  <P3D>
4690 NEXT L                                             <5G6>
4700 :                                                  <C1H>
4710 REM *** ZEICHNE RAHMEN ***                         <5SN>
4720 :                                                  <D1L>
4730 H=N*10+55: PRINT#2,0                               <MOG>
4740 PRINT#1,"D",0,H: PRINT#1,"D",479,H: PRINT#1,
     "D",479,0: PRINT#1,"D",0,0                         <JP5>
4750 PRINT#1,"M",0,H-40: PRINT#1,"D",479,H-40           <E86>
4760 PRINT#1,"M",60,H: PRINT#1,"D",60,0                 <JVR>
4770 PRINT#1,"M",275,H: PRINT#1,"D",275,0               <MF7>
4780 PRINT#4: PRINT#4: PRINT#4                          <0K2>
4790 :                                                  <8H3>
4800 REM *** AUSGABE DER BERECHNETEN KOEFFIZIENTE
     N ***                                              <248>
4810 :                                                  <3GF>
```

Listing 1. (Fortsetzung)

```
4820 PRINT#6,0: PRINT#2,1: PRINT#1,"M",90,0: PRIN
     T#4,A$(1): PRINT#4: PRINT#2,2                 <S0K>
4830 FOR L=1 TO NG                                 <ANP>
4840 IF SP>1 THEN N$=CHR$(L+64): ELSE N$="A"+MID$
     (STR$(L-1),2)                                 <6LB>
4850 IF ABS(C(L))>1E-4 THEN X=INT(C(L)*1E6+.5)/1E
     6: ELSE X=C(L)                                <JAJ>
4860 PRINT#1,"M",100,0: PRINT#4,N$;"=";X           <AB5>
4870 NEXT L                                        <L26>
4880 PRINT#6,1                                     <1LL>
4890 :                                             <7GV>
4900 REM *** ZEICHNE RAHMEN ***                    <2KA>
4910 :                                             <2GB>
4920 PRINT#2,0: H=NG*10+35                         <FBE>
4930 PRINT#1,"D",0,H: PRINT#1,"D",479,H: PRINT#1,
     "D",479,0: PRINT#1,"D",0,0                    <5VT>
4940 PRINT#4: PRINT#4: PRINT#4                     <2K2>
4950 ZZ=1: GOTO 1410                               <7KF>
4960 :                                             <105>
4970 REM *** FUNKTIONENDEFINITIONEN ***            <3CD>
4980 :                                             <60P>
4990 REM *** POLYNOM 1. GRADES ***                 <0S3>
5000 :                                             <915>
5010 DEF FN A(X)=C(1)+C(2)*X: M=1: RETURN          <2K6>
5020 :                                             <811>
5030 REM *** POLYNOM 2. GRADES ***                 <7SV>
5040 :                                             <B1D>
5050 DEF FN A(X)=C(1)+C(2)*X+C(3)*X*X: M=2: RETUR
     N                                             <2SR>
5060 :                                             <A19>
5070 REM *** POLYNOM 3. GRADES ***                 <6SR>
5080 :                                             <D1L>
5090 DEF FN A(X)=C(1)+C(2)*X+C(3)*X*X+C(4)*X*X*X:
      M=3: RETURN                                  <P6V>
5100 :                                             <811>
5110 REM *** POLYNOM 4. GRADES ***                 <7SV>
5120 :                                             <915>
5130 DEF FN A(X)=C(1)+C(2)*X+C(3)*X*X+C(4)*X↑3+C(
     5)*X↑4: M=4: RETURN                           <F6V>
5140 :                                             <A19>
5150 REM *** Y=A*EXP(B*X) ***                      <6SR>
5160 :                                             <B1D>
5170 FOR L=1 TO N                                  <B5D>
5180 Y(L)=LOG(YP(L))/LOG(10)                       <688>
5190 NEXT L                                        <LQ6>
5200 :                                             <B1D>
5210 M=1: RETURN                                   <F05>
5220 C(1)=10↑C(1): C(2)=C(2)*LOG(10)               <E1U>
5230 DEF FN A(X)=C(1)*EXP(C(2)*X)                  <22M>
5240 RETURN                                        <9K7>
5250 :                                             <9H7>
5260 REM *** Y=A*X↑B ***                           <7KU>
5270 :                                             <8H3>
5280 FOR L=1 TO N                                  <CLD>
5290 X(L)=LOG(XP(L))/LOG(10): Y(L)=LOG(YP(L))/LOG
     (10)                                          <TGA>
5300 NEXT L                                        <TN6>
5310 :                                             <AHB>
5320 M=1: RETURN                                   <E05>
5330 C(1)=10↑C(1)                                  <0B1>
5340 DEF FN A(X)=C(1)*X↑C(2)                       <KC2>
5350 RETURN                                        <PI7>
5360 :                                             <915>
5370 REM *** Y=1/(A+B*X) ***                       <7CT>
5380 :                                             <E1P>
5390 FOR L=1 TO N                                  <D5D>
5400 Y(L)=1/YP(L): X(L)=XP(L)                      <DQS>
5410 NEXT L                                        <506>
5420 :                                             <C1H>
5430 M=1: RETURN                                   <105>
5440 DEF FN A(X)=1/(C(1)+C(2)*X)                   <EKE>
5450 RETURN                                        <PE7>
5460 :                                             <E1P>
5470 REM *** Y=X/(A+B*X) ***                       <4SJ>
5480 :                                             <915>
5490 FOR L=1 TO N                                  <A5D>
5500 Y(L)=XP(L)/YP(L): X(L)=XP(L)                  <AK6>
5510 NEXT L                                        <LQ6>
5520 :                                             <D1L>
5530 M=1: RETURN                                   <305>
5540 DEF FN A(X)=X/(C(1)+C(2)*X)                   <4S3>
5550 RETURN                                        <PA7>
5560 :                                             <F1T>
5570 REM *** Y=A*LN(B*X) ***                       <4CH>
5580 :                                             <811>
5590 FOR L=1 TO N                                  <B5D>
5600 X(L)=LOG(XP(L)): Y(L)=YP(L)                   <ME0>
5610 NEXT L                                        <5S6>
5620 M=1: RETURN                                   <405>
5630 :                                             <EHR>
5640 DEF FN A(X)=C(1)*LOG(C(2)*X)                  <EMM>
5650 A=C(1): C(1)=C(2): C(2)=EXP(A/C(2))           <SJG>
5660 RETURN                                        <907>
5670 :                                             <CHJ>
5680 REM *** AUSGABE AUF BILDSCHIRM ***            <640>
5690 :                                             <BHF>
5700 GOSUB 6500 : REM *** EINGABE                  <M7B>
5710 :                                             <EHR>
5720 COLOR 0,1,0: COLOR 4,1,0: GRAPHIC 1,1         <4LC>
5730 :                                             <FHV>
5740 XS=(B-A)/320: YS=(D-C)/200                    <OE0>
5750 Y0=INT(ABS(A/XS)): X0=INT(ABS(D/YS))          <3B9>
5760 IF X0>199 THEN X0=X0-1                        <B8C>
5770 IF Y0>319 THEN Y0=Y0-1                        <OD0>
5780 :                                             <A19>
5790 FOR I=0 TO 22: FOR J=0 TO 22: DRAW 1,15.95*J
     ,9.95*I: NEXT J,I                             <9BM>
```

```
5800 :                                             <105>
5810 Z=2: COLOR 1,2,7                              <GGC>
5820 DRAW 1,0,X0 TO 319,X0                         <VF9>
5830 DRAW 1,Y0,0 TO Y0,199                         <DAF>
5840 :                                             <30D>
5850 FOR J=0 TO 22                                 <B7N>
5860 IF Z=3 THEN Z=2: ELSE Z=3                     <UDH>
5870 DRAW 1,J*15.95,X0-Z TO 15.95*J,X0+Z           <MAJ>
5880 DRAW 1,Y0-Z,9.95*J TO Y0+Z,9.95*J             <3RO>
5890 NEXT J                                        <5G5>
5900 :                                             <001>
5910 X=X0/8-1: IF X<1 THEN X=1                     <4GE>
5920 Y=Y0/8+1: IF Y>35 THEN Y=Y-4                  <URN>
5930 CHAR 1,0,X,STR$(A): CHAR 1,36,X,STR$(B)       <0DF>
5940 CHAR 1,Y,0,STR$(D): CHAR 1,Y,24,STR$(C)       <MJG>
5950 TRAP 7350                                     <LE7>
5960 :                                             <30D>
5970 REM *** KENNZEICHUNG DER MESSWERTE ***        <2C9>
5980 :                                             <40H>
5990 IF PO=0 THEN 6060                             <48G>
6000 FOR I=1 TO N                                  <ONT>
6010 X=INT((XP(I)-A)/XS+.5)                        <BTT>
6020 Y=199-INT((YP(I)-C)/YS+.5)                    <2KV>
6030 IF X > 0 AND Y > 0 THEN CIRCLE 1,X,Y,2        <LL0>
6040 NEXT I                                        <DD4>
6050 :                                             <DHN>
6060 FOR V=1 TO ZZ                                 <1P3>
6070 IF V=1 THEN J=FN A(A): F=3                    <Q3Q>
6080 IF V=2 THEN J=FN B(A): F=6                    <IIP>
6090 IF V=3 THEN J=FN C(A): F=7                    <AHP>
6100 COLOR 1,F,7                                   <SQ1>
6110 CHAR 1,0,V-1,A$(V),1                          <DBP>
6120 Y=199-INT((J-C)/YS+.5)                        <5S7>
6130 IF Y>0 THEN LOCATE 0,Y: ELSE LOCATE 0,0       <U66>
6140 :                                             <C1H>
6150 FOR XW=0 TO 319                               <9DV>
6160 TRAP 6260                                     <JU2>
6170 I=A+XW*XS                                     <O38>
6180 IF V=1 THEN J=FN A(I)                         <65H>
6190 IF V=2 THEN J=FN B(I)                         <3LV>
6200 IF V=3 THEN J=FN C(I)                         <755>
6210 Y=199-INT((J-C)/YS+.5)                        <LM7>
6220 IF Y < 0 AND Z < 0 THEN LOCATE XW,0: GOTO 62
     60                                            <2UI>
6223 IF (Y < 0 AND Z > 199) OR (Y > 199 AND Z < 0
     ) THEN 6260                                   <ATF>
6225 IF Y < 0 AND Z > 0 THEN DRAW 1 TO XW,0 : GOT
     O 6260                                        <717>
6230 IF Y < 0 AND Z > 0 THEN DRAW 1 TO XW,0 : GOT
     O 6260                                        <01R>
6240 IF (Y < 0 AND Z > 199) OR (Y > 199 AND Z < 0
     ) THEN 6260                                   <IIF>
6250 DRAW 1 TO XW,Y                                <T8I>
6260 Z=Y: NEXT XW: NEXT V: TRAP                    <TRQ>
6270 :                                             <EHR>
6280 COLOR 4,7,0: COLOR 1,8,7                      <8FF>
6290 GET KEY A$: IF ASC(A$)<>13 THEN 6290          <QES>
6300 GRAPHIC 0: COLOR 0,7,0                        <FVT>
6310 IF PO=1 THEN 3930                             <5KN>
6320 GOTO 1000                                     <7OK>
6330 END                                           <IIE>
6340 REM *** EINGABEROUTINE ***                    <4KI>
6350 A$=""                                         <BDH>
6360 PRINT "?";: GET KEY B$ : PRINT CHR$(157);     <STC>
6370 W=ASC(B$)                                     <L2T>
6380 IF W>44 AND W<58 THEN PRINT B$;: GOTO 6430    <KUM>
6390 IF W=13 AND LEN(A$)>0 THEN 6450               <H1L>
6400 IF W<>20 THEN 6440                            <EVV>
6410 PRINT CHR$(157);"? ";CHR$(157);CHR$(157);     <3JL>
6420 A$=LEFT$(A$,LEN(A$)-1): GOTO 6440             <EHO>
6430 A$=A$+B$                                      <00E>
6440 GOTO 6360                                     <208>
6450 W=VAL(A$): PRINT " ";                         <MAU>
6460 RETURN                                        <9G7>
6470 :                                             <BH3>
6480 REM *** UP-EINGABE ***                        <44G>
6490 :                                             <FHV>
6500 PRINT CHR$(147): PRINT                        <L6F>
6510 PRINT "DARSTELLUNGSBEREICH DER FUNKTION AUF"
     : PRINT                                       <VR7>
6520 PRINT "DER X-ACHSE VON XU= ";: GOSUB 6340: A
     =W: PRINT : PRINT                             <B5C>
6530 IF A>0 THEN PRINT "XU MUSS KLEINER GLEICH 0
     SEIN !": PRINT : GOTO 6520                    <T4R>
6540 PRINT TAB(12);"BIS XO= ";: GOSUB 6340: B=W:
     PRINT : PRINT                                 <B9J>
6550 IF B<0 THEN PRINT "XO MUSS GROESSER GLEICH 0
      SEIN !": PRINT : GOTO 6540                   <F7L>
6560 PRINT "DARSTELLUNGSBEREICH DER FUNKTION AUF"
     : PRINT                                       <SB9>
6570 PRINT "DER Y-ACHSE VON YU= ";: GOSUB 6340: C
     =W: PRINT : PRINT                             <QFC>
6580 IF C>0 THEN PRINT "YU MUSS KLEINER GLEICH 0
     SEIN !": PRINT : GOTO 6570                    <60S>
6590 PRINT TAB(12);"BIS YO= ";: GOSUB 6340: D=W:
     PRINT                                         <730>
6600 IF D<0 THEN PRINT "YO MUSS GROESSER GLEICH 0
      SEIN !": PRINT : GOTO 6590                   <G3P>
6610 RETURN                                        <PM7>
6620 :                                             <811>
6630 REM *** COMPUTER WAEHLT FUNKTION ***          <7SV>
6640 PRINT CHR$(147);"BERECHNUNG IM FAST-MODUS (J
     /N) ?"                                        <E7P>
6650 GET KEY A$                                    <20D>
6660 IF A$="J" THEN FAST                           <1I6>
6670 :                                             <AHB>
6680 FG=1                                          <URJ>
```

```
6690 FOR SP=1 TO 9                                      <6HL>
6700 FE=0                                               <NQN>
6710 ON SP GOSUB 5010,5050,5090,5130,5150,5260,53
     70,5470,5570                                       <244>
6720 GOTO 3310                                          <60R>
6730 F(SP)=S: IF FE=1 THEN F(SP)=-1                     <MH9>
6740 PRINT : PRINT CHR$(155);"FEHLER = ";F(SP)          <AIG>
6750 :                                                  <AHB>
6760 FE=0: GOSUB 7020                                   <3AM>
6770 :                                                  <BHF>
6780 NEXT SP                                            <MFR>
6790 :                                                  <CHJ>
6800 SLOW                                               <E9G>
6810 PRINT CHR$(147);CHR$(158);CHR$(18);"{2SPACE}
     FUNKTION";TAB(27);"FEHLER"                         <97S>
6820 F1=1E10: F2=0: FG=2                                <0RF>
6830 FOR SP=1 TO 9                                      <GHL>
6840 PRINT : PRINT FK$(SP);TAB(25);F(SP)                <KTR>
6850 IF F1=F(SP) AND LEN(FK$(F2))>LEN(FK$(SP)) TH
     EN F2=SP                                           <Q4M>
6860 IF F1>F(SP) AND F(SP)<>-1 THEN F1=F(SP): F2=
     SP                                                 <3QP>
6870 NEXT SP                                            <7DR>
6880 PRINT CHR$(155): SP=F2: FE=0                       <8FK>
6890 PRINT "AUSGEWAEHLTE FUNKTION IST": PRINT CHR
     $(18);FK$(SP)                                      <389>
6900 FOR L=1 TO N: X(L)=XP(L): Y(L)=YP(L): NEXT L       <DB6>
6910 PRINT : PRINT CHR$(18);"WEITER MIT >RETURN<"
     ;                                                  <7P3>
6920 GET KEY A$: IF ASC(A$)<>13 THEN 6920               <CU6>
6930 PRINT " OK";CHR$(158)                              <428>
6940 GOTO 3280                                          <28G>
6950 :                                                  <4GJ>
6960 REM *** FEHLERROUTINE ***                          <144>
6970 :                                                  <5GN>
6980 PRINT CHR$(147);"LOESUNG NICHT VORHANDEN FUE
     R": PRINT : PRINT CHR$(18);FK$(SP)                 <FA5>
6990 FE=1                                               <3K7>
7000 RESUME NEXT                                        <OUQ>
7010 :                                                  <DHN>
7020 REM *** SUMMEN 0 SETZEN ***                        <5KM>
7030 :                                                  <CHJ>
7040 SY=0                                               <KUR>
7050 FOR L=1 TO 4: XY(L)=0: NEXT                        <GQ2>
7060 FOR L=1 TO 8: SX(L)=0: NEXT                        <R82>
7070 RETURN                                             <PA7>
7080 :                                                  <915>
7090 REM *** UEBERPRUEFUNG DER MESSWERTE ***            <7CT>
7100 :                                                  <C1H>
7110 PRINT CHR$(147);CHR$(18);CHR$(158);"{2SPACE}
     MESSWERTE{2SPACE}"                                 <U1F>
7120 FOR I=1 TO N                                       <QNT>
7130 PRINT                                              <3SU>
7140 PRINT I;".{2SPACE}";"X= ";XP(I);TAB(23);"Y=
     ";YP(I)                                            <268>
7150 NEXT I                                             <LA4>
7160 PRINT CHR$(155)                                    <AVD>
7170 PRINT "ALLE MESSWERTE KORREKT ? (J/N)              <5B0>
7180 GET KEY A$                                         <30D>
7190 IF A$="J" THEN 7300                                <SPI>
7200 IF A$="N" THEN 7220                                <U1T>
7210 GOTO 7180                                          <206>
7220 PRINT                                              <RRU>
7230 PRINT "NUMMER DES FALSCHEN MESSWERTES ";           <8LP>
7240 GOSUB 6340: I=W: PRINT                             <FJ9>
7250 PRINT : PRINT I;". X= ";: GOSUB 6340: X=W          <U3B>
7260 PRINT TAB(23);"Y= ";: GOSUB 6340: Y=W: PRINT
                                                        <SK2>
7270 XP(I)=X: X(I)=X: YP(I)=Y: Y(I)=Y                   <F4L>
7280 GOTO 7110                                          <20F>
7290 :                                                  <BHF>
7300 PRINT : PRINT CHR$(18);"ZURUECK INS MENUE MI
     T >RETURN<"                                        <I36>
7310 GET KEY A$: IF ASC(A$)<>13 THEN 7310               <P2H>
7320 GOTO 4200                                          <204>
7330 :                                                  <FHV>
7340 REM *** FEHLERMELDUNG ***                          <5KM>
7350 GRAPHIC 0                                          <SJP>
7360 PRINT CHR$(147);"ANFANGSWERT HAT FEHLER VERU
     RSACHT."                                           <6P5>
7370 PRINT "BERECHNUNGEN WERDEN NICHT FORTGEFUEHR
     T"                                                 <183>
7380 PRINT : PRINT CHR$(18);"WEITER MIT >RETURN<"       <R2V>
7390 GET KEY A$: IF ASC(A$)<>13 THEN 7390               <RIR>
7400 IF PO=1 THEN 4200                                  <N8S>
7410 GOTO 1000                                          <5GK>
7420 :                                                  <811>
8000 REM *** EINGEFUEGTE FUNKTIONEN ***                 <A58>
8010 DEF FN A(X)=SIN(X)+SIN(20*X)/5                     <427>
8020 DEF FN B(X)=SIN(20*X)/5                            <SGV>
8030 DEF FN C(X)=SIN(X)                                 <I49>
8040 A$(1)="Y=SIN(X)+SIN(20*X)/5"                       <T82>
8050 A$(2)="Y=SIN(20*X)/5"                              <7RU>
8060 A$(3)="Y=SIN(X)"                                   <MMH>
8070 ZZ= 3                                              <0S8>
8100 RETURN                                             <A07>
```

Listing 1. Das Programm »Approxi 128« berechnet die mathematische Funktion zu Ihren Versuchsergebnissen (Schluß)



For 128 only

Spiele, die den vollen Speicher des C128 ausnutzen, gibt es wenig. Zwei dieser Programme haben wir unter die Lupe genommen.

Eine häufige Beschwerde der C128-Besitzer lautet: »Für meinen Computer gibt es keine Spiele!«. Es gibt zwar tausende von Programmen für den C64, die auch auf dem C128 funktionieren; Programme, die speziell auf den C128 zurechtgeschnitten sind und auch den vollen Speicher ausnutzen, sind dagegen sehr dünn gesät. Zu den wenigen Programmen, die nur mit 128 KByte Arbeitsspeicher lauffähig sind, gehören einige Adventures der Firma Infocom.

Was ist eigentlich ein Adventure? Ein Adventure läßt sich am besten als ein Roman im Computer beschreiben. Der Unterschied zu einem gedruckten Roman besteht darin, daß Sie die Rolle der Hauptperson übernehmen und aktiv in die Handlung eingreifen. Zuerst beschreibt das Adventure die Situation, in der Sie sich gerade befinden. Sie können dann mit einem englischen Satz dem Programm mitteilen, was Sie als nächstes tun wollen. Die Handlung des Spiels ist dabei immer mit zahlreichen logischen Rätseln verknüpft. Ein Beispiel: Vor Ihnen steht ein Löwe, der Sie unbedingt fressen möchte. Sie haben nun aber ein Stück Fleisch, was Sie vorher mit einem Schlafmittel versehen haben, bei sich. Mit der Eingabe »Give the meat to the lion« geben Sie dem Löwen das Fleisch. Dieser frißt es begierig und schläft ein. Hätten Sie dem Löwen beispielsweise die Konservendose gegeben, hätte das Programm vielleicht geantwortet, daß der Löwe kurz an der Dose schnüffelt und Ihnen dann herzhaft ins Bein beißt...

Zwei besonders komplexe und nicht gerade einfache Vertreter dieser Spielegattung stellen wir Ihnen nun vor. Beide Programme sind auch von der Handlung her sehr ungewöhnlich, geht es hier doch nicht um das Erschlagen von Drachen und das Retten von Prinzessinnen, sondern um den Krieg mit Formularen und Behörden und die Bedrohung durch die Atombombe.

Bureaucracy

Es war einmal, vor nicht allzu langer Zeit, ein Mann, der innerhalb Londons umziehen wollte. Er benachrichtigte jedermann über seine neue Adresse, auch seine Bank; ja, er ging sogar selbst dorthin und füllte persönlich ein entsprechendes Formular aus. Er hatte vor, danach glücklich bis ans Ende seiner Tage in der neuen Wohnung zu leben.

Doch nur wenige Tage später mußte er feststellen, daß seine Bank seine Kreditkarte für ungültig erklärt hatte. Eine neue Kreditkarte war rechtzeitig an ihn geschickt worden – aber an die alte Adresse.

Mehrere Wochen lang versuchte der Mann vergeblich, seine neue Adresse bei der Bank durchzusetzen. Er sprach mit vielen Bankangestellten, füllte viele weitere Formulare aus, beantragte eine zweite Kreditkarte, aber nichts von alledem half.

Der Mann, der mehrere Wochen lang ohne einen Penny leben mußte, ist Douglas Adams, Autor der berühmten »Hitchhikers Guide to the Galaxy«-Serie (Per Anhalter durch die Galaxis), die er schon zu einem Adventure verwandelt hatte. Nach knapp zwei Jahren hat Douglas seine Wut über diese Episode ausgelassen und ein witziges Adventure daraus gemacht. Ach, übrigens: Douglas bekam auch ein Entschuldigungsschreiben vom Bankdirektor. Man sandte es natürlich an seine alte Adresse.

Bei »Bureaucracy« ist die Ausgangslage ähnlich verzweifelt. Sie haben einen neuen Job, eine neue Wohnung und ähnliche Probleme mit Ihrer Bank. Heute nachmittag sollen Sie nach Paris fliegen, um an einem Lehrgang teilzunehmen und auch etwas auszuspannen. Doch die wichtigsten Dinge, das Flug-Ticket, Geld sowie Ihre neue Kreditkarte sind bei Ihrer alten Adresse gelandet. Und da der neue Besitzer Ihres ehemaligen Hauses etwas wütend auf Sie ist (Sie hätten ihm doch nicht die Sache mit den Termiten und Küchenschaben verschweigen sollen), hat er Ihre wichtige

```
 Back Room                                           Blood Pressure: 134/87
curtains, no furniture. A room to go bughouse in, really. Another room is
visible to the west, an                        side.
                    The word "Boysenberry," and
>west               the symbol of a partially
Back Room           digested purple berry, are
                    trademarks of the Boysenberry
You're in the back room Business Engines Corporation. e room, at present
suitable for lining wit                          f the walls, but likely
to be pretty impressive and upwardly mobile once the removals men have sorted
out their little problem. The exit leads east to the living room.

You see a combination telephone/answering machine and a table here. On the
table you see a hacksaw, an address book, a small case, your Boysenberry
computer, a letter and your passport.

>examine the computer
This is the remarkable Boysenberry laptop computer, made by a subsidiary of
your old employers, the Deep Thought Corporation of America. There are no
operating controls of any sort, not even an on/off switch. All you see are a
tiny screen, a keyboard, a modular jack and a slot for program cartridges.

The doorbell is ringing.

>go to the front door then open it
```

Kampf gegen Formulare und Bürokraten bei Bureaucracy

```
                              Lancaster Walk

An impressive sculpture of a horse and rider dominates this bustling
intersection. The Walk continues north and south; lesser paths curve off in
many directions.

A broad field of grass, meticulously manicured, extends to the east. Beyond it
you can see the Long Water glittering between the trees.

There's a notice here.

>read the notice
The words Do Not Walk On The Grass are sternly printed on the notice.

>go east
As your feet touch the grass you sense a strange motion around you. Looking
down, you watch with horror as the grass begins to ripple and writhe with
vegetable indignance!

Angry green stalks whip around your legs, pull you to the ground and drag you,
kicking and screaming, back to the paved surface of the Lancaster Walk.

A chorus of offended little voices subsides as you regain your footing.

>examine the grass
```

Bei Trinity müssen Sie einen Atomkrieg verhindern

Post einfach nicht angenommen, so daß sich diese jetzt irgendwo auf unserem Globus befinden könnte.

Gleich zu Beginn des Programms erwartet Sie eine unangenehme Überraschung. Trotz eingelegter Original-Diskette erscheint auf dem Bildschirm eine Beschwerde, daß dies wohl eine Raubkopie sei. Aber das ist ja alles nicht so schlimm, denn wenn man ein Formular zum Antrag einer vorläufigen Nutzungslizenz ausfüllt, erlaubt Ihnen das Programm, weiterzuspielen. Das Formular erscheint prompt auf dem Bildschirm und will nicht nur Namen und Adresse, sondern auch die Namen der letzten Freundinnen/Freunde, die am meisten gehaßte Farbe sowie den vorletzten Arbeitgeber wissen. Natürlich merkt sich Bureaucracy, was Sie da eingeben, und verwendet diese Daten im weiteren Verlauf des Spiels.

Diese Episode ist nur der Anfang einer Odyssee durch den Dschungel der Bürokratie, die allerdings auch durch den Dschungel einiger afrikanischer Kleinstaaten führt. Sie streifen durch Fast-Food-Restaurants, Reisebüros, Lama-Farmen, Flughäfen, die Häuser Ihrer Nachbarn und sogar durch einige Banken, um endlich an Ihr verdientes Geld zu kommen.

Auf dem Bildschirm rechts oben, wo man üblicherweise bei einem Adventure Punkte und Spielzüge sieht, wird Ihr aktueller Blutdruck angezeigt. Immer, wenn Sie etwas falsch machen, steigt der Blutdruck, ansonsten fällt er langsam. Wenn Ihr Blutdruck einen gewissen Wert übersteigt, spielt Ihr Kreislauf nicht mehr mit - endlich mal ein Adventure mit natürlicher Todesursache!

Bureaucracy ist ein selbst für Infocom-Verhältnisse ungewöhnliches Adventure. Die immer wieder auftauchenden Formulare, die Punkte-Zählung mit dem Blutdruck, die zahlreichen Gags und Anspielungen auf unser tägliches Leben, all dies macht Bureaucracy zu einem Adventure-Erlebnis, das man so schnell sicher nicht vergißt.

Trinity

In nicht allzuferner Zukunft ist die politische Lage zwischen Ost und West sehr gespannt. An den Grenzen vieler europäischer Staaten werden verstärkt Truppen stationiert. So brauchen Sie gerade in diesen harten Zeiten ein wenig Ablenkung und Zerstreuung.

Sie übernehmen die Rolle eines Touristen, der sich gerade auf einer 599-Dollar-Pauschalreise durch Europa befindet. Die letzte Station vor dem Heimflug ist London. Plötzlich heulen Sirenen auf. Der lange befürchtete dritte Weltkrieg hat begonnen. Aber bevor feindliche Atomraketen Ihre Umgebung in Schutt und Asche legen können, öffnet sich ein Dimensionstor, mit dem Sie in verschiedene Epochen der jüngeren Geschichte reisen können. Jedes Ereignis, daß Sie aufsuchen werden, hat mit dem Thema Atombombe zu tun. Ihre einzige Chance, den dritten Weltkrieg wieder rückgängig zu machen, besteht darin, einen Weg zu finden, um die erste Atombomben-Explosion der Welt zu verhindern. Diese fand am 16. Juli 1945 auf einem amerikanischen Testgelände in der Wüste von Nevada statt. Der Codename des Atombombenprojekts: Trinity.

Mit »Trinity« liegt das erste Adventure von Infocom vor, das geschickt Fantasie-Elemente mit realen Gegebenheiten und Orten verquickt.

Trinity ist nicht ohne politische Brisanz und gehört zu den wenigen Spielen, bei denen der Autor eine klare Aussage vertritt. Dazu kommen eine umfangreiche Handlung (der Sourcecode von Trinity benötigt etwa 1,4 MByte) und eine große Zahl von kniffligen Problemen.

Zu den Beilagen bei Trinity gehört der fast schon zynische Comic: »Die illustrierte Geschichte der Atombombe« der in bester Walt-Disney- und Superman-Manier die Vorteile dieser Wunderwaffe aufzählt. Außerdem findet man einen Bastelbogen für eine Sonnenuhr, die auch in der Handlung des Spiels eine Schlüsselrolle inne hat.

Beide Programme werden nur auf Diskette ausgeliefert. Obwohl sie den Speicher von 128 KByte ausnutzen, muß doch ständig von Diskette nachgeladen werden, was oft zu unangenehmen Wartezeiten führt. Wie Sie vielleicht schon vermutet haben, sind beide Programme nur in englischer Sprache erhältlich. (bs)

Info: Activision, Postfach 76 06 80, 2000 Hamburg 76



Wallstreet im Wohnzimmer

Werden Sie zum Aktienspekulanten. Mit der realistischen Börsensimulation »Börse 128« erleben Sie auf spielerische Art und Weise hautnah die Vorgänge an einer Aktienbörse.

Bei »Börse 128« können maximal neun Mitspieler ihr Glück an der Börse versuchen. Die Dauer des Spiels wird am Anfang festgelegt und kann maximal 9999 Spieltage betragen. Jeder Spieler erhält 25000 Mark Startkapital. Ziel des Spieles ist, sein Vermögen innerhalb der Spielzeit durch geschickten Kauf und Verkauf von Aktien zu vermehren. Das heißt, die Aktien zu einem niedrigen Kurs einzukaufen, aber zu einem hohen Kurs zu verkaufen. Von 24 Firmen stehen Aktien zur Verfügung, deren Kurse sich (spiel-)täglich verändern. Von Zeit zu Zeit erscheint eine Telex-Nachricht über die wirtschaftliche Situation einer Firma auf dem Bildschirm. Aufgrund dieser Nachricht wird die Aktienkursentwicklung der Firma entscheidend beeinflußt.

Boerse 128

Dies ist der 15. Tag an der Boerse

Firma	Kurs	Firma	Kurs
Apple	95	Nixdorf	177
Acorn Computer	98	Olivetti	106
Atari	189	Oric	89
Brother	157	Osborne	127
Casio	72	Panasonoic	98
Citizen	114	Philips	97
Commodore	85	Sanyo	108
Digital Research	115	Schneider	105
Epson	99	Sharp	112
Hewlett Packard	107	Sinclair	96
I.B.M.	108	Spectravideo	100
NEC	95	Texas Instrument	88

Bild 1. Das umfangreiche Auswahlmenü von »Börse 128«

Auch Angebot und Nachfrage beeinflussen das Geschehen und die Kursentwicklung an der Börse. Von jeder Firma stehen 2000 Aktien zum Verkauf zu Verfügung. Fällt der Aktienkurs einer Firma unter die Zehn-Mark-Grenze, dann bedeutet das, daß die Firma bankrott ist und alle Aktien dieser Firma zugleich wertlos sind.

Kauft ein Spieler Aktien an, so werden pro Aktie 2% an Gebühren berechnet. Beim Verkauf fallen diese 2% ebenfalls an. Läßt sich ein Spieler die Kosten für ein Aktienpaket berechnen, kauft aber dieses anschließend nicht, so werden ihm diese Gebühren ebenfalls angelastet. Letzteres gilt auch für den Verkauf von Aktien.

Hüten Sie sich vor Fehlkäufen

Der Spieler kann sich während des Spiels die eigene Kapitalentwicklung oder die Aktienkursentwicklung einer bestimmten Firma in Form eines Blockdiagramms anzeigen lassen. Den Maßstab für die Kapital- und Aktienkursachse errechnet das Programm bei jedem Diagrammaufruf neu. So wird die größtmögliche Auflösung und eine optimale Bildschirmausnutzung gewährleistet.

Am Ende eines Spiels wird ermittelt, welcher Spieler am

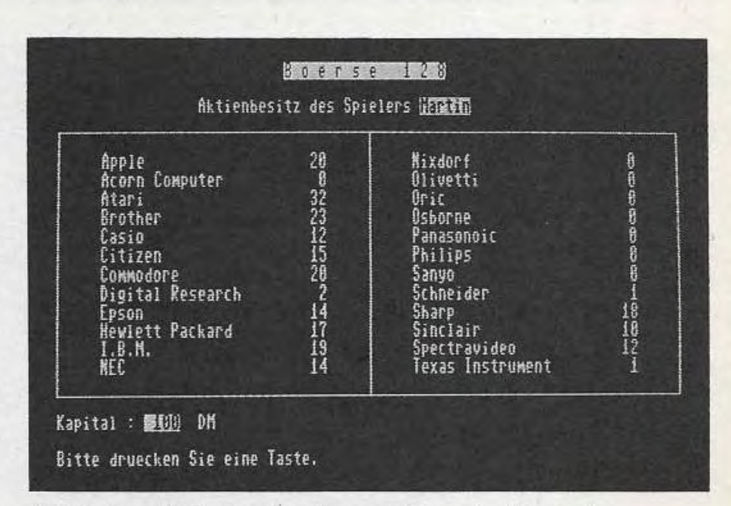


Bild 2. Der Aktienbesitz eines Spielers im Überblick

erfolgreichsten war. Dazu wird anhand der Aktienkurse des letzten Spieltages der Wert der Aktien, die der Spieler besitzt, errechnet und sein Bargeldvermögen dazu addiert. Wer am meisten Kapital besitzt, hat das Spiel gewonnen.

Bitte geben Sie Börse 128 (Listing 1) mit dem Checksummer 128 ein (beachten Sie dazu die Eingabehinweise auf Seite 158). Nach dem Start mit »RUN« gelangen Sie zum Titelbild und können sich auf Wunsch eine kurze Spielanleitung auf dem Bildschirm ausgeben lassen. Anschließend kann ein eventuell vorhandener alter Spielstand geladen werden.

Neun »Börsenmakler« gleichzeitig

Geben Sie nun die Anzahl der Spieler (maximal 9), der Spieltage und die Namen der Spieler ein, und Sie gelangen ins Auswahlmenü (Bild 1).

Wollen Sie Aktien kaufen, so drücken Sie Taste <1>. Mit <F1> und <F3> oder mit den Cursortasten wählt man nun die Firma an, deren Aktien man zu kaufen wünscht, mit <F5> bestätigt man die Wahl. Falls Sie diesen Menüpunkt versehentlich angewählt haben, gelangen Sie mit <F7> wieder ins Auswahlmenü. Nun können Sie die Anzahl der

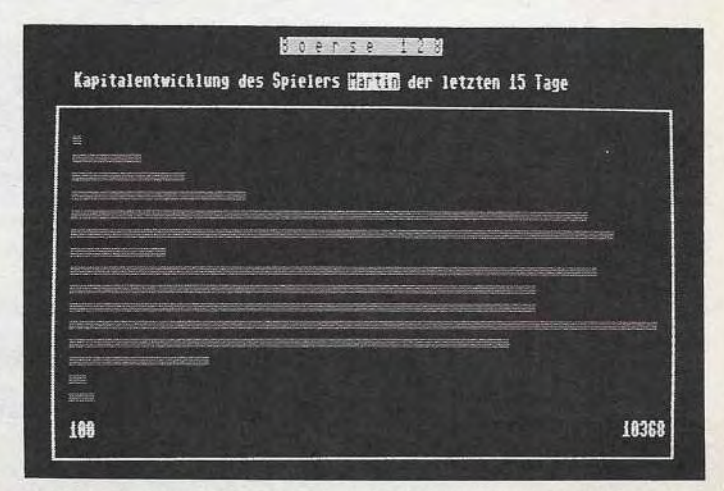


Bild 3. Der Kapitalbesitz eines Spielers ist oft großen und manchmal unerwarteten Schwankungen unterworfen

gewünschten Aktien eingeben; die maximal mögliche Menge wird vorher angezeigt. Nach Ausgabe der Kaufsumme können Sie nun engültig entscheiden, ob Sie den Kauf tätigen wollen oder nicht. Die Bearbeitungsgebühr von 2% wird jedoch in jedem Fall verrechnet. Noch ein Hinweis: Sollten Sie versehentlich mit dem Cursor außerhalb des Eingabefeldes geraten, so positionieren Sie diesen wieder im Eingabefeld und fügen am Ende der Eingabe einen Doppelpunkt an.

Mit <2> aktiviert man den Verkauf von Aktien. Die Bedienung ist hier ähnlich wie bei »Ankauf«.

Durch Taste <3> wird Ihr Besitz an Aktien angezeigt (Bild 2). Der Bildschirmaufbau ist ähnlich dem Auswahlmenü, nur daß hinter dem Firmennamen keine Kurse stehen, sondern die Anzahl der Aktien, die der Spieler von dieser Firma besitzt. Mit einem beliebigen Tastendruck, erfolgt ein Rücksprung in das Auswahlmenü.

Durch <5> wird eine ansprechende Balkengrafik der Kapitalentwicklung des jeweiligen Spielers über die letzten 15 (Spiel-)Tage ausgegeben (Bild 3).

Der Computer unterstützt Sie

Bei <6> erhalten Sie eine Grafik der letzten 15 Aktienkurse einer bestimmten Firma. Diese wählt man wieder mit den Cursortasten oder <F1>/<F3> aus und bestätigt die Wahl mit <F5>. Ein Beispiel einer solchen Grafik sehen Sie in Bild 4. Mit einem beliebigen Tastendruck gelangt man wieder in das Auswahlmenü zurück. Wenn ein Spieler alle seine Aktivitäten beendet hat, wählt man <4> und der nächste Spieler kommt an die Reihe.



Bild 4. Die simulierten Kursschwankungen einer Aktie

Haben alle Spieler ihre Aktionen beendet, bricht ein neuer Börsentag an.

Will man ein Spiel unterbrechen und zu späterer Zeit fortsetzen, kann der aktuelle Spielstand vom Auswahlmenü aus mit <F8> jederzeit auf Diskette gespeichert werden. In Tabelle 1 und 2 finden Sie noch einmal eine kurze Zusammenfassung aller Tastenbelegungen.

Für Interessierte folgt nun eine kurze Dokumentation des Listings von »Börse 128«. Das Programm ist in Basic V7.0 geschrieben und arbeitet im 80 Zeichen-Modus. Es teilt sich in ein Hauptprogramm und mehrere Unterprogramme, die in Tabelle 3 ausführlich beschrieben sind.

Eine Liste aller Variablen und deren Bedeutung sehen Sie in Tabelle 4. Sämtliche Variablen werden im Programmkopf aufgerufen. Die Formel für die Berechnung der neuen Kurse ist so gewählt, daß sich die Schwankungen der Aktienkurse in vernünftigen Grenzen halten.

(Stefan Sure/Thomas Lipp)

Taste	Funktion
1	Kaufen von Aktien
2	Verkaufen von Aktien
3	Anzeige des Aktienbesitzes
4	Eingabe beenden, nächster Spieler
5	Anzeige der Kapitalentwicklung
6	Anzeige der Aktienkursentwicklung
F8	Speichern des Spielstandes

Tabelle 1. Tasten im Auswahlmenü mit ihren Funktionen

Taste	Funktion
F1	nächste Firma
F3	letzte Firma
F5	Firma auswählen
F7	Abbruch der Aktion

Tabelle 2. Funktionstastenbelegung bei An- und Verkauf

Zeilennr.	Unterprogramm
930	Einlesen von Variablen
1230	Anzeige des Auswahlmenüs für Spieleraktionen
1360	Anzeige der Tageskurse
1550	Anzeigen der aktuellen Nachricht
1720	Durchführen der Spieleraktion Kaufen
2370	Durchführen der Spieleraktion Verkaufen
2920	Anzeige des Auswahlmenüs Kaufen/Verkaufen
3020	Anzeige des Aktienbesitzes eines Spielers
3210	Anzeige des Diagramms Besitzenwicklung
3620	Anzeige des Diagramms Aktienentwicklung
4050	Auswahl einer Firma
4380	Anzeige wenn Firma bankrott geht
4490	Titelgrafik, Spielanleitung, Datas einlesen
5410	Anzeige der Abschlußtabelle
5750	Speichern des Spielstandes auf Diskette
6150	Laden des Spielstandes von Diskette

Tabelle 3. Die Unterprogramme von »Börse 128«

Name	Feldgröße	Aufgabe im Programm
AF%		Anzahl der Firmen
AS%		Anzahl der Spieler
AT%		Anzahl der Spieltage
TA%		Aktueller Spieltag
F$		Tastaturabfrage
Z		Zufallszahl
WK%		Anzahl zu kaufender/verkaufender Aktien
WF%		Von welcher Firma kaufen/verkaufen
HI		Schleifenzähler Hauptprogramm
VP		Verkaufspreis/Rückgabepreis
MA%		Max. Anzahl der Aktien, die man kaufen kann
IZ%		Input-Zeile
IS%		Input-Spalte
IL%		Input-Länge
IN%		Input-Type
IP$		Input-String
SK	AS%	Spielerkapital
SA%	AS%,24	Aktien des Spielers von einer Firma
SN$	AS%	Spielername
AK%	24	Aktienkurs einer Firma
NF$	24	Firmenname
FM%	32	Fernschreibermeldung
MU%	32	Multiplikator für Aktienkurs
FI%	32	Firma auf die sich Meldung bezieht
MA%	24	Aktueller Multiplikator
AA%	24	Zur Verfügung stehende Aktien einer Firma
BE	15,AS%	Kapitalentwicklung eines Spielers
AE%	15,24	Aktienentwicklung einer Firma
MI	15	Sortierfeld Kapital/Kurse
I-I3		Schleifenzähler
C1$-C8$		Steuerzeichen

Tabelle 4. Die Liste aller verwendeten Variablen

```
100 REM                                          <IIA>
110 REM ************************************** <IQB>
120 REM *    PROGRAMM : BOERSE 128           * <MGN>
130 REM ************************************** <IA9>
140 REM *    AUTOR    : STEFAN SURE          * <EJS>
150 REM *               GARTENSTRASSE 2      * <JQF>
160 REM *               5882 MEINERZHAGEN 1  * <J2C>
170 REM *               TEL.: 02354 - 4499   * <GDQ>
180 REM ************************************** <GI2>
190 REM *               AUGUST 1986          * <GQ3>
200 REM ************************************** <J2C>
210 REM                                          <JAD>
220 REM * VARIABLENLISTE *                       <JIE>
230 REM                                          <JQF>
240 FAST                                         <PJ3>
250 I=0: I1=0: I2=0: I3=0: HI=0: TA%=0: WF$=""   <G30>
260 AF%=24: AS%=0: AT%=0: Z=0: WA%=0: F$=""      <K8V>
270 IZ%=0: IS%=0: IL%=0: IN%=0: IP$=""           <DEJ>
280 C1$=CHR$(27): C2$="@": C3$=CHR$(147): C4$=CHR
    $(145)                                       <T3N>
290 C5$=CHR$(81): C6$=CHR$(19): C7$=CHR$(17): C8$
    =CHR$(157)                                   <4G8>
300 DIM AE%(15,AF%),AK%(AF%),MI(15),MA%(AF%),NF$(
    AF%),AA%(AF%),MU%(32),FM$(32),FI%(32)        <IR3>
310 GOSUB 4490 : REM VORBEREITUNG                <AFE>
320 GOSUB 5290 : REM DATAS EINLESEN              <DFU>
330 REM                                          <JAD>
340 REM * HAUPTPROGRAMM *                        <IIA>
350 REM                                          <IQB>
360 PRINT C6$+C7$+C1$+C2$                        <9CF>
370 DO                                           <HM7>
380 TA%=TA%+1                                    <NVD>
390 GOSUB 1360 : REM KURSANZEIGE                 <8U4>
400 FOR HI=1 TO AS%                              <JTA>
410 DO                                           <PN7>
420 GOSUB 1230 : REM AUSWAHLMENUE SPIELER        <EEA>
430 GET KEY F$                                   <LKA>
440 IF F$="4" THEN EXIT                          <SLS>
450 IF F$=CHR$(138) THEN GOSUB 5750 : END        <23P>
460 IF F$<"1" OR F$>"6" THEN 430                 <URN>
470 ON VAL(F$) GOSUB 1720,2370,3020,100,3210,3610 <2AU>
480 LOOP                                         <J29>
490 IF TA%<=14 THEN BEGIN                        <S7K>
500 BE(TA%,HI)=SK(HI): BE(TA%+1,HI)=SK(HI)       <2RK>
510 BEND : ELSE BEGIN                            <FA3>
520 FOR I=1 TO 13                                <O0V>
530 BE(I,HI)=BE(I+1,HI)                          <799>
540 NEXT                                         <9D0>
550 BE(14,HI)=SK(HI): BE(15,HI)=SK(HI)           <NSH>
560 BEND                                         <ATE>
570 NEXT                                         <5CG>
580 CHAR ,4,19,""                                <622>
590 PRINT C1$+C2$+C4$                            <AVU>
600 COLOR 5,7                                    <LOD>
610 PRINT TAB(4)"DIE BOERSE HAT GESCHLOSSEN."    <DHI>
620 PRINT                                        <TFU>
630 IF TA%=AT% THEN EXIT                         <G2J>
640 PRINT TAB(4)"BEREITEN SIE SICH AUF EINEN NEUE
    N TAG VOR."                                  <J6D>
650 COLOR 5,6                                    <KKD>
660 REM                                          <GI2>
670 REM * NEUE KURSE BERECHNEN *                 <GQ3>
680 REM                                          <J2C>
690 FOR I=1 TO AF%                               <BTO>
700 AK%(I)=AK%(I)*(.93+.155*RND(2)+MA%(I)*.0032) <SIN>
710 IF AK%(I)<10 THEN GOSUB 4380 : REM FIRMENBANK
    ROTT                                         <HA0>
720 MA%(I)=MA%(I)/(RND(1)*1.8+.9)                <97V>
730 IF ABS(MA%(I))<1 THEN MA%(I)=0               <CR7>
740 NEXT                                         <PF0>
750 Z=RND(0)*10+1                                <NIT>
760 IF INT(Z)>5 AND INT(Z)<8 THEN GOSUB 1550 : RE
    M AKTUELLE NACHRICHT                         <OGI>
770 IF TA%<=14 THEN BEGIN                        <K6K>
780 FOR I=1 TO AF%                               <DTO>
790 AE%(TA%+1,I)=AK%(I)                          <RGP>
800 NEXT                                         <H20>
810 BEND : ELSE BEGIN                            <EU5>
820 FOR I1=1 TO AF%                              <PDU>
830 FOR I=1 TO 14                                <3SS>
840 AE%(I,I1)=AE%(I+1,I1)                        <525>
850 NEXT                                         <50G>
860 AE%(15,I1)=AK%(I1)                           <1NQ>
870 NEXT                                         <D1G>
880 BEND                                         <QNE>
890 LOOP                                         <NEP>
900 GOSUB 5410 : REM ABSCHLUSSBILANZ             <PFM>
910 PRINT C3$+CHR$(142)                          <3U3>
920 END                                          <916>
930 REM                                          <MAP>
940 REM * INPUT SUBROUTINE *                     <NIU>
950 REM                                          <NQV>
960 OPEN 1,0,1                                   <ALC>
970 CHAR ,IS%,IZ%," {LEFT}"                      <EVG>
980 INPUT#1,IP$                                  <OGQ>
990 CLOSE 1                                      <H6N>
1000 IF LEN(IP$)>IL% THEN GOSUB 1190: GOTO 960   <F5I>
1010 IP$=LEFT$(IP$,IL%)                          <ODH>
1020 IF IN%=1 THEN BEGIN                         <FLM>
1030 FOR I=1 TO LEN(IP$)                         <LL4>
1040 IF ASC(MID$(IP$,I,1))>57 OR ASC(MID$(IP$,I,1
     ))<48 AND ASC(MID$(IP$,I,1))<>46 THEN GOSUB
     1190: GOTO 960                              <4LD>
1050 NEXT                                        <MAG>
1060 RETURN                                      <807>
1070 BEND                                        <LQU>
1080 IF IN%=2 THEN BEGIN                         <DTS>
1090 FOR I=1 TO IL%                              <ST4>
1100 IF ASC(MID$(IP$,I,1))>57 OR ASC(MID$(IP$,I,1
     ))<48 AND ASC(MID$(IP$,I,1))<>46 THEN GOSUB
     1190: GOTO 960                              <SLD>
1110 NEXT                                        <E9G>
1120 RETURN                                      <8K7>
1130 BEND                                        <DPU>
1140 IF IN%=3 THEN BEGIN                         <E5I>
1150 IF LEN(IP$)<IL% THEN GOSUB 1190: GOTO 960   <1CA>
1160 RETURN                                      <8S7>
1170 BEND                                        <TRU>
1180 RETURN                                      <807>
1190 FOR I=0 TO LEN(IP$)                         <8HC>
1200 CHAR ,IS%+I,IZ%," "                         <D1H>
1210 NEXT                                        <MAG>
1220 RETURN                                      <807>
1230 REM                                         <2SB>
1240 REM * AUSWAHLMENUE SPIELER *                <34C>
1250 REM                                         <3CD>
1260 CHAR ,4,19,""                               <L2E>
1270 PRINT C1$+C2$+C4$                           <RPQ>
1280 COLOR 5,9                                   <54A>
1290 PRINT USING "{4SPACE}SPIELER{2SPACE}: {RVSON
     }################{RVOFF,6SPACE}KAPITAL{2SPACE
     }: {2SPACE,RVSON}########{RVOFF,2SPACE}DM";SN
     $(HI),SK(HI)                                <1AT>
1300 PRINT                                       <R3U>
1310 PRINT TAB(4)"{RVSON} 1 {RVOFF}.....KAUFEN{9S
     PACE,RVSON} 3 {RVOFF}.....AKTIENBESITZ{3SPAC
     E,RVSON} 5 {RVOFF}.....BESITZENTWICKLUNG"   <D9F>
1320 PRINT                                       <B1U>
1330 PRINT TAB(4)"{RVSON} 2 {RVOFF}.....VERKAUFEN
     {6SPACE,RVSON} 4 {RVOFF}.....WEITER{9SPACE,R
     VSON} 6 {RVOFF}.....AKTIENENTWICKLUNG"+C4$  <HGH>
1340 COLOR 5,6                                   <78D>
1350 RETURN                                      <0I7>
1360 REM                                         <34C>
1370 REM * KURSANZEIGE *                         <3CD>
1380 REM                                         <0K2>
1390 PRINT C6$+C7$                               <0T6>
1400 COLOR 5,10                                  <C84>
1410 IF TA%<AT% THEN BEGIN                       <QE5>
1420 PRINT TAB(22)"DIES IST DER "TA%;C8$". TAG AN
      DER BOERSE"                                <UD1>
1430 BEND : ELSE BEGIN                           <33I>
1440 PRINT TAB(21)"DIES IST DER LETZTE TAG AN DER
      BOERSE"                                    <7HP>
1450 BEND                                        <TVU>
1460 COLOR 5,6                                   <48D>
1470 PRINT TAB(4)"CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
     CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC"  <MQB>
1480 PRINT TAB(4)"G" TAB(40)"G" TAB(75)"H"       <SSK>
1490 FOR I=1 TO 12                               <LSS>
1500 PRINT USING "{4SPACE}G{4SPACE}##############
     ##{5SPACE}#####{5SPACE}H{4SPACE}############
     ####{5SPACE}#####{4SPACE}H";NF$(I),AK%(I),NF
     $(I+12),AK%(I+12)                           <PP7>
1510 NEXT                                        <EDG>
1520 PRINT TAB(4)"G" TAB(40)"G" TAB(75)"H"       <KTK>
1530 PRINT TAB(4)"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
     TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT"  <P3M>
1540 RETURN                                      <887>
1550 REM                                         <0S3>
1560 REM * AKTUELLE NACHRICHT AUSGEBEN *         <040>
1570 REM                                         <0C1>
1580 Z=INT(RND(1)*32+1)                          <3C1>
1590 IF LEFT$(NF$(FI%(Z)),1)="*" THEN RETURN     <8SM>
1600 CHAR ,4,19,""                               <D6E>
1610 PRINT C1$+C2$                               <LMJ>
1620 COLOR 5,8                                   <5GA>
1630 CHAR ,4,19,"*{2SPACE}AKTUELLE NACHRICHTEN{2S
     PACE}*{2SPACE}AKTUELLE NACHRICHT{2SPACE}*{2S
     PACE}AKTUELLE NACHRICHT{2SPACE}*"           <J4L>
1640 CHAR ,4,21,"*{2SPACE}"+FM$(Z)+"{2SPACE}*",1 <AP9>
1650 CHAR ,4,23,"*{2SPACE}AKTUELLE NACHRICHTEN{2S
     PACE}*{2SPACE}AKTUELLE NACHRICHT{2SPACE}*{2S
     PACE}AKTUELLE NACHRICHT{2SPACE}*"           <77L>
1660 COLOR 5,6                                   <58D>
1670 SLEEP 3                                     <067>
1680 MA%(FI%(Z))=MU%(Z)                          <FIL>
1690 CHAR ,4,19,""                               <92E>
1700 PRINT C1$+C2$                               <PMJ>
1710 RETURN                                      <0A7>
1720 REM                                         <040>
1730 REM * KAUFEN *                              <0C1>
1740 REM                                         <1K6>
1750 GOSUB 2920 : REM AUSWAHLMENUE KAUFEN/VERKAUF
     EN                                          <67V>
1760 GOSUB 4050 : REM AUSWAHL FIRMA              <3N5>
1770 WF%=I1                                      <69V>
1780 IF F$=CHR$(137) THEN RETURN                 <9QQ>
1790 IF AA%(WF%)=0 THEN BEGIN                    <8U0>
1800 CHAR ,4,21,""                               <JBN>
1810 PRINT C1$+C2$                               <5QI>
1820 CHAR ,4,21,"ES STEHEN KEINE AKTIEN DER FIRMA
     {2SPACE,RVSON}"+NF$(WF%)+"{RVOFF,2SPACE}ZU V
     ERFUEGUNG."                                 <EV3>
1830 SLEEP 3                                     <PM7>
1840 RETURN                                      <9S7>
1850 BEND                                        <TJU>
1860 CHAR ,4,21,""                               <BBN>
1870 PRINT C1$+C2$                               <TQI>
1880 CHAR ,4,21,"SIE KAUFEN AKTIEN DER FIRMA{2SPA
     CE,RVSON}"+NF$(WF%)+"{RVOFF}.{3SPACE}AKTIENK
     URS :{2SPACE,RVSON}"+STR$(AK%(WF%))+"{RVOFF}
      DM"                                        <0VM>
1890 MA%=INT(SK(HI)/(AK%(WF%)*1.02))             <LD4>
```

```
1900 IF AA%(WF%)<MA% THEN MA%=AA%(WF%)                <DBB>
1910 CHAR ,4,23,"WIEVIEL AKTIEN WOLLEN SIE KAUFEN
     [MAX {RVSON}"+STR$(MA%)+"{RVOFF} ] ?"             <3PS>
1920 IZ%=23: IS%=52: IL%=4: IN%=1: GOSUB 930: WA%
     =VAL(IP$)                                         <5R9>
1930 IF WA%>AA%(WF%) THEN BEGIN                        <JCP>
1940 CHAR ,4,21,""                                     <BBN>
1950 PRINT C1$+C2$                                     <TQI>
1960 CHAR ,4,21,"ES STEHEN NICHT GENUG AKTIEN DER
     FIRMA{2SPACE,RVSON}"+NF$(WF%)+"{RVOFF,2SPAC
     E}ZU VERFUEGUNG."                                 <S1H>
1970 CHAR ,4,23,"BITTE TASTE DRUECKEN."                <NCC>
1980 GET KEY F$                                        <00A>
1990 RETURN                                            <P27>
2000 BEND                                              <PVE>
2010 IF WA%*AK%(WF%)*1.02>INT(SK(HI)) THEN BEGIN       <AA7>
2020 CHAR ,4,21,""                                     <B7M>
2030 PRINT C1$+C2$                                     <TMJ>
2040 CHAR ,4,21,"FUER DIESES GESCHAEFT HABEN SIE
     NICHT GENUG KAPITAL.",1                           <9EK>
2050 CHAR ,4,23,"BITTE TASTE DRUECKEN.{25SPACE}"       <9PC>
2060 GET KEY F$                                        <40A>
2070 RETURN                                            <027>
2080 BEND                                              <PRE>
2090 VP%=WA%*AK%(WF%)*1.02                             <0E8>
2100 CHAR ,4,19,""                                     <56E>
2110 PRINT C1$+C2$+C4$                                 <OPM>
2120 PRINT USING "{4SPACE}AKTIENWERT{24SPACE}:{6S
     PACE}####{2SPACE}DM";AK%(WF%)                     <N65>
2130 PRINT USING "{4SPACE}AKTIENPREIS INCL. 2% GE
     BUEHREN{4SPACE}:{5SPACE}#####{2SPACE}DM";INT
     (AK%(WF%)*1.02)                                   <GGN>
2140 PRINT USING "{4SPACE}GESAMTPREIS FUER ####{2
     SPACE}AKTIEN{5SPACE}:{3SPACE}#######{2SPACE}
     DM";WA%,VP%                                       <3LL>
2150 PRINT USING "{4SPACE}VERBLEIBENDES KAPITAL{1
     3SPACE}:{3SPACE}#######{2SPACE}DM";INT(SK(HI
     ))-VP%                                            <I6V>
2160 CHAR ,4,23,"WOLLEN SIE DAS GESCHAEFT ABSCHLI
     ESSEN ? [J/N]",1                                  <1Q4>
2170 DO                                                <TFN>
2180 GET KEY F$                                        <70A>
2190 IF F$="J" THEN BEGIN                              <R4C>
2200 SA%(HI,WF%)=SA%(HI,WF%)+WA%                       <PQS>
2210 SK(HI)=INT(SK(HI)-VP%)                            <4IK>
2220 MA%(WF%)=MA%(WF%)+2                               <708>
2230 AA%(WF%)=AA%(WF%)-WA%                             <0B3>
2240 IF TA%<=14 THEN BE(TA%,HI)=SK(HI): RETURN         <VFM>
2250 BE(15,HI)=SK(HI)                                  <FFV>
2260 RETURN                                            <887>
2270 BEND                                              <LUU>
2280 IF F$="N" THEN BEGIN                              <QK2>
2290 CHAR ,4,19,""                                     <P2E>
2300 PRINT C1$+C2$+C4$                                 <PHJ>
2310 PRINT TAB(4)"DIE BANK BERECHNET IHNEN EINE B
     EARBEITUNGSGEBUEHR VON ";INT(WA%*AK%(WF%)*.0
     15);" DM."                                        <SKV>
2320 SK(HI)=INT(SK(HI)-WA%*AK%(WF%)*.015)              <FPP>
2330 SLEEP 3                                           <027>
2340 RETURN                                            <887>
2350 BEND                                              <LUU>
2360 LOOP                                              <G69>
2370 REM                                               <0C1>
2380 REM * VERKAUFEN *                                 <3KE>
2390 REM                                               <3SF>
2400 GOSUB 2920 : REM AUSWAHLMENUE KAUFEN/VERKAUF
     EN                                                <97V>
2410 GOSUB 4080 : REM AUSWAHL FIRMA                    <87V>
2420 IF F$=CHR$(137) THEN RETURN                       <9QQ>
2430 WF%=I1                                            <6LV>
2440 IF SA%(HI,WF%)=0 THEN BEGIN                       <K5G>
2450 CHAR ,4,21,""                                     <N3M>
2460 PRINT C1$+C2$                                     <9IJ>
2470 CHAR ,4,21,"SIE BESITZEN KEINE AKTIEN DER FI
     RMA {RVSON}"+NF$(WF%)+"{RVOFF}."                  <N79>
2480 SLEEP 3                                           <887>
2490 RETURN                                            <0E7>
2500 BEND                                              <HQE>
2510 CHAR ,4,21,""                                     <F3M>
2520 PRINT C1$+C2$                                     <HIJ>
2530 CHAR ,4,21,"SIE VERKAUFEN AKTIEN DER FIRMA {
     RVSON}"+NF$(WF%)+"{RVOFF}"+". SIE BESITZEN {
     RVSON}"+STR$(SA%(HI,WF%))+"{RVOFF} AKTIEN."       <TB2>
2540 CHAR ,4,23,"WIEVIEL AKTIEN WOLLEN SIE VERKAU
     FEN ?"                                            <PE3>
2550 IZ%=23: IS%=42: IL%=4: IN%=1: GOSUB 930: WA%
     =VAL(IP$)                                         <6N4>
2560 IF SA%(HI,WF%)<WA% THEN BEGIN                     <B7K>
2570 CHAR ,4,21,""                                     <N3M>
2580 PRINT C1$+C2$                                     <PMJ>
2590 CHAR ,4,21,"FUER DIESES GESCHAEFT HABEN SIE
     NICHT GENUG AKTIEN.",1                            <PKG>
2600 CHAR ,4,23,"BITTE TASTE DRUECKEN.{25SPACE}"       <0PC>
2610 GET KEY F$                                        <60A>
2620 RETURN                                            <8G7>
2630 BEND                                              <5OU>
2640 VP%=WA%*AK%(WF%)*.98                              <G8L>
2650 CHAR ,4,19,""                                     <92E>
2660 PRINT C1$+C2$+C4$                                 <QHV>
2670 PRINT USING "{4SPACE}AKTIENWERT{28SPACE}:{6S
     PACE}####{2SPACE}DM";AK%(WF%)                     <TOT>
2680 PRINT USING "{4SPACE}RUECKNAHMEPREIS ABZUEGL
     . 2% GEBUEHREN :{5SPACE}#####{2SPACE}DM";INT
     (AK%(WF%)*.98)                                    <JBP>
2690 PRINT USING "{4SPACE}RUECKNAHMEPREIS FUER ##
     ##{2SPACE}AKTIEN{5SPACE}:{3SPACE}#######{2SP
     ACE}DM";WA%,VP%                                   <6MF>
2700 CHAR ,4,23,"WOLLEN SIE DIE AKTIEN VERKAUFEN
     ? [J/N]",1                                        <TF2>
2710 DO                                                <NEV>
2720 GET KEY F$                                        <68A>
2730 IF F$="J" THEN BEGIN                              <D40>
2740 SA%(HI,WF%)=SA%(HI,WF%)-WA%                       <3Q4>
2750 SK(HI)=INT(SK(HI)+VP%)                            <8F4>
2760 MA%(WF%)=MA%(WF%)-2                               <QJU>
2770 AA%(WF%)=AA%(WF%)+WA%                             <M5R>
2780 IF TA%<=14 THEN BE(TA%,HI)=SK(HI): RETURN         <F9M>
2790 BE(15,HI)=SK(HI)                                  <EVP>
2800 RETURN                                            <9C7>
2810 BEND                                              <TNU>
2820 IF F$="N" THEN BEGIN                              <LLU>
2830 CHAR ,4,19,""                                     <1EF>
2840 PRINT C1$+C2$+C4$                                 <T11>
2850 PRINT TAB(4)"DIE BANK BERECHNET IHNEN EINE B
     EARBEITUNGSGEBUEHR VON ";INT(WA%*AK%(WF%)*.0
     12);" DM."                                        <L9R>
2860 SK(HI)=INT(SK(HI)-WA%*AK%(WF%)*.012)              <68U>
2870 SLEEP 3                                           <P67>
2880 RETURN                                            <9S7>
2890 BEND                                              <TJU>
2900 LOOP                                              <OF9>
2910 RETURN                                            <PA7>
2920 REM                                               <44G>
2930 REM * AUSWAHLMENUE KAUFEN/VERKAUFEN               <4CH>
2940 REM                                               <5KM>
2950 CHAR ,4,21,""                                     <VFN>
2960 PRINT C1$+C2$                                     <1UI>
2970 COLOR 5,9                                         <0CA>
2980 CHAR ,4,21,"{RVSON} F1 {RVOFF}.....VORWAERTS
     {15SPACE,RVSON} F5 {RVOFF}.....AUSWAHL"           <8UQ>
2990 CHAR ,4,23,"{RVSON} F3 {RVOFF}.....RUECKWAER
     TS"                                               <2QV>
2992 IF F$="6" THEN 3000                               <R11>
2994 CHAR ,24,23,"{13SPACE,RVSON} F7 {RVOFF}.....
     ABBRUCH"                                          <JT9>
3000 COLOR 5,6                                         <5OD>
3010 RETURN                                            <O67>
3020 REM                                               <1K6>
3030 REM * AKTIENBESITZ *                              <1S7>
3040 REM                                               <040>
3050 PRINT C6$+C7$+C1$+C2$                             <E8V>
3060 PRINT TAB(20)"AKTIENBESITZ DES SPIELERS {RVS
     ON}"+SN$(HI)+"{RVOFF}"                            <2UN>
3070 PRINT TAB(4)"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"         <MQB>
3080 PRINT TAB(4)"G" TAB(40)"G" TAB(75)"H"             <SSK>
3090 FOR I=1 TO 12                                     <LSS>
3100 PRINT USING "{4SPACE}G{4SPACE}###############
     #{5SPACE}#####{5SPACE}H{4SPACE}#############
     ####{5SPACE}#####{4SPACE}H";NF$(I),SA%(HI,I)
     ,NF$(I+12),SA%(HI,I+12)                           <8DN>
3110 NEXT                                              <EDG>
3120 PRINT TAB(4)"G" TAB(40)"G" TAB(75)"H"             <KTK>
3130 PRINT TAB(4)"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
     TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT"         <P3M>
3140 PRINT TAB(4)"KAPITAL : {RVSON}";INT(SK(HI));
     "{RVOFF} DM"                                      <SDH>
3150 PRINT                                             <JAU>
3160 PRINT TAB(4)"BITTE DRUECKEN SIE EINE TASTE."      <K37>
3170 GET KEY F$                                        <50A>
3180 PRINT C6$+C7$+C1$+C2$                             <IFF>
3190 GOSUB 1360 : REM KURSANZEIGE                      <067>
3200 RETURN                                            <8C7>
3210 REM                                               <0C1>
3220 REM * BESITZENTWICKLING *                         <0K2>
3230 REM                                               <0S3>
3240 PRINT C6$+C7$+C1$+C2$                             <QAF>
3250 PRINT TAB(6)"KAPITALENTWICKLUNG DES SPIELERS
      {RVSON}"+SN$(HI)+"{RVOFF} DER LETZTEN 15 TA
     GE"                                               <6KE>
3260 PRINT TAB(4)"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"         <UVB>
3270 FOR I=1 TO 19                                     <UGP>
3280 PRINT TAB(4)"G" TAB(75)"H"                        <6DR>
3290 NEXT                                              <MAG>
3300 PRINT TAB(4)"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
     TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT"         <14M>
3310 FOR I=1 TO 15                                     <R0V>
3320 MI(I)=BE(I,HI)                                    <DHJ>
3330 NEXT                                              <MEG>
3340 FOR I1=1 TO 14                                    <52P>
3350 FOR I=1 TO 14                                     <V4V>
3360 IF MI(I)>MI(I+1) THEN BEGIN                       <GH2>
3370 I2=MI(I)                                          <04H>
3380 MI(I)=MI(I+1)                                     <TJL>
3390 MI(I+1)=I2                                        <K8K>
3400 BEND                                              <9PE>
3410 NEXT                                              <68G>
3420 NEXT                                              <A90>
3430 I3=(MI(15)-MI(1))/68                              <CQA>
3440 CHAR ,5,21,""                                     <B2M>
3450 PRINT USING "#######";STR$(MI(1))                 <7PO>
3460 CHAR ,68,21,""                                    <MND>
3470 PRINT USING ">######";STR$(MI(15))                <U1N>
3480 COLOR 5,15                                        <CS6>
3490 FOR I1=1 TO 15
                                                       <36P>
3500 IF BE(I1,HI)>0 THEN BEGIN                         <TG5>
3510 FOR I=6 TO  5+INT((BE(I1,HI)-MI(1))/I3)           <GDN>
3520 CHAR ,I,4+I1,"T"                                  <N7M>
```

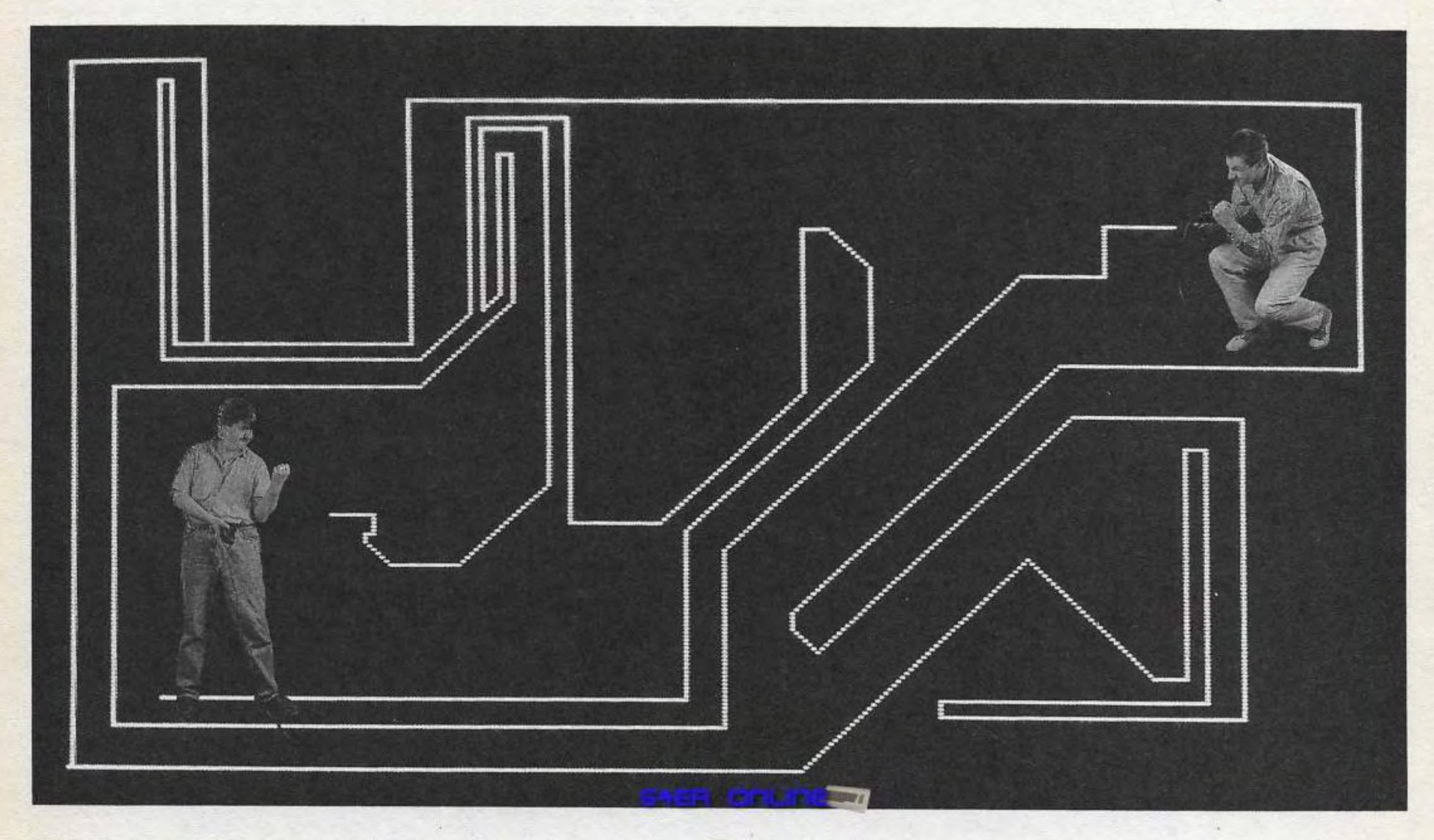
Listing 1. »Börse 128«, bitte mit dem Checksummer 128 (Seite 158) eingeben

```
3530 NEXT                                                  <68G>
3540 BEND                                                  <HQE>
3550 NEXT                                                  <UBG>
3560 COLOR 5,6                                             <60D>
3570 GET KEY F$                                            <70A>
3580 PRINT C6$+C7$+C1$+C2$                                 <IBF>
3590 GOSUB 1360                                            <NND>
3600 RETURN                                                <8S7>
3610 REM                                                   <2C9>
3620 REM *AKTIENENTWICKLUNG *                              <2KA>
3630 REM                                                   <2SB>
3640 CHAR ,4,19,"VON WELCHER FIRMA MOECHTEN SIE D
     IE BILANZEN DER LETZTEN 15 TAGE SEHEN?"               <P4C>
3650 GOSUB 2920                                            <LNH>
3660 GOSUB 4050                                            <NFS>
3670 WF%=I1                                                <6TV>
3680 PRINT C6$+C7$+C1$+C2$                                 <A8F>
3690 PRINT TAB(8)"AKTIENENTWICKLUNG DER FIRMA {RV
     SON}"+NF$(WF%)+"{RVOFF} DER LETZTEN 15 TAGE"          <H2D>
3700 PRINT TAB(4)"BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
     BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"           <UJB>
3710 FOR I=1 TO 19                                         <IGP>
3720 PRINT TAB(4)"G" TAB(75)"H"                            <6DR>
3730 NEXT                                                  <MAG>
3740 PRINT TAB(4)"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
     TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT"           <18M>
3750 FOR I=1 TO 15                                         <N0V>
3760 MI(I)=AE%(I,WF%)                                      <MM7>
3770 NEXT                                                  <68G>
3780 FOR I1=1 TO 14                                        <D2P>
3790 FOR I=1 TO 14                                         <R4V>
3800 IF MI(I)>MI(I+1) THEN BEGIN                           <KH2>
3810 I2=MI(I)                                              <44H>
3820 MI(I)=MI(I+1)                                         <TFM>
3830 MI(I+1)=I2                                            <88K>
3840 BEND                                                  <PNE>
3850 NEXT                                                  <M6G>
3860 NEXT                                                  <Q70>
3870 I3=(MI(15)-MI(1))/68                                  <SKA>
3880 CHAR ,5,21,""                                         <RAN>
3890 PRINT USING "####";STR$(MI(1))                        <GTT>
3900 CHAR ,70,21,""                                        <61E>
3910 PRINT USING ">###";STR$(MI(15))                       <TQB>
3920 COLOR 5,15                                            <4S6>
3930 FOR I1=1 TO 15                                        <32P>
3940 IF AE%(I1,WF%)>0 THEN BEGIN                           <T4D>
3950 FOR I=6 TO  5+INT((AE%(I1,WF%)-MI(1))/I3)             <RU5>
3960 CHAR ,I,4+I1,"T"                                      <KN0>
3970 NEXT                                                  <M6G>
3980 BEND                                                  <1GE>
3990 NEXT                                                  <E1G>
4000 COLOR 5,6                                             <20D>
4010 GET KEY F$                                            <30A>
4020 PRINT C6$+C7$+C1$+C2$                                 <25F>
4030 GOSUB 1360                                            <FND>
4040 RETURN                                                <9K7>
4050 REM                                                   <7CT>
4060 REM * FIRMEN AUSWAHL *                                <7KU>
4070 REM                                                   <7SV>
4080 I1=1: I2=9: I3=0                                      <FH6>
4090 CHAR ,I2,5,NF$(1),1                                   <UIR>
4100 DO UNTIL F$=CHR$(135) OR F$=CHR$(137)                 <08U>
4110 GET KEY F$                                            <3GA>
4120 IF F$=C4$ THEN BEGIN
                                                           <DDB>
4130 I1=I1-1                                               <RDB>
4140 IF I1<1 THEN BEGIN                                    <LJ4>
4150 CHAR ,I2,I1+5,NF$(I1+I3+1)                            <104>
4160 I1=12                                                 <JQU>
4170 IF I2=9 THEN I2=45: I3=12: ELSE I2=9: I3=0            <A01>
4180 BEND : ELSE BEGIN                                     <IBN>
4190 CHAR ,I2,I1+5,NF$(I1+I3+1)                            <704>
4200 BEND                                                  <9HE>
4210 CHAR ,I2,I1+4,NF$(I1+I3),1                            <18H>
4220 BEND                                                  <1GE>
4230 IF F$=C7$ THEN BEGIN                                  <FD5>
4240 I1=I1+1                                               <QPD>
4250 IF I1>12 THEN BEGIN                                   <992>
4260 CHAR ,I2,I1+3,NF$(I1+I3-1)                            <0G8>
4270 I1=1                                                  <MQJ>
4280 IF I2=9 THEN I2=45: I3=12: ELSE I2=9: I3=0            <0Q9>
4290 BEND : ELSE BEGIN                                     <27M>
4300 CHAR ,I2,I1+3,NF$(I1+I3-1)                            <QG8>
4310 BEND                                                  <5GU>
4320 CHAR ,I2,I1+4,NF$(I1+I3),1                            <10H>
4330 BEND                                                  <DHU>
4340 LOOP                                                  <0B9>
4350 CHAR ,I2,I1+4,NF$(I1+I3)                              <RP9>
4360 I1=I1+I3                                              <DPQ>
4370 RETURN                                                <PU7>
4380 REM                                                   <5KM>
4390 REM * FIRMENBANKROTT *                                <5SN>
4400 REM                                                   <44G>
4410 CHAR ,4,23,"DIE FIRMA {RVSON}"+NF$(I)+"{RVOF
     F} HAT BANKROTT GEMACHT!"                             <3U5>
4420 SLEEP 2                                               <987>
4430 NF$(I)="****************"                             <C2R>
4440 AK%(I)=0                                              <R73>
4450 FOR I1=1 TO AS%                                       <317>
4460 SA%(I1,I)=0                                           <GK7>
4470 NEXT                                                  <E5G>
4480 RETURN                                                <9S7>
4490 REM                                                   <6CP>
4500 REM * VORBEREITUNG *                                  <4KI>
4510 REM                                                   <4SJ>
4520 COLOR 6,1: COLOR 5,8                                  <14N>
4530 PRINT CHR$(14)+C3$;                                   <7FR>
4540 PRINT TAB(30)"{RVSON}B O E R S E{3SPACE}1 2
     8{RVOFF}"                                             <J8K>
4550 COLOR 5,6                                             <10D>
4560 CHAR ,0,4,""                                          <0IV>
4570 PRINT TAB(16)"BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
     BBBBBBBBBBBBBBBBB"                                    <13T>
4580 PRINT TAB(16)"G" TAB(62)"H"                           <VA8>
4590 PRINT TAB(16)"G{2SPACE}HERZLICH WILLKOMMEN A
     N DER 'BOERSE 128'!{3SPACE}H"                         <T2D>
4600 PRINT TAB(16)"G" TAB(62)"H"                           <0A8>
4610 PRINT TAB(16)"G{3SPACE}DIESES SPIEL IST EIN
     WIRTSCHAFTSSPIEL.{4SPACE}H"                           <1PQ>
4620 PRINT TAB(16)"G" TAB(62)"H"                           <PA8>
4630 PRINT TAB(16)"G GESCHRIEBEN IM AUGUST 1986 V
     ON STEFAN SURE. H"                                    <HLC>
4640 PRINT TAB(16)"G" TAB(62)"H"                           <QA8>
4650 PRINT TAB(16)"G{2SPACE}WUENSCHEN SIE EINE SP
     IELANLEITUNG ? [J/N]{2SPACE}H"                        <75D>
4660 PRINT TAB(16)"G" TAB(62)"H"                           <RA8>
4670 PRINT TAB(16)"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
     TTTTTTTTTTTTTTTTT"                                    <H01>
4680 DO                                                    <4CJ>
4690 GET KEY F$                                            <20A>
4700 IF F$="J" THEN BEGIN                                  <LLM>
4710 PRINT C6$+C7$+C1$+C2$                                 <M3V>
4720 CHAR ,0,3,""                                          <7I3>
4730 PRINT TAB(12)"BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
     BBBBBBBBBBBBBBBBBBBBBBBBB"                            <SVK>
4740 PRINT TAB(12)"G" TAB(66)"H"                           <JF8>
4750 PRINT TAB(12)"G AN DIESEM WIRTSCHAFTSSPIEL K
     OENNEN BIS ZU 9 SPIELER H"                            <0EJ>
4760 PRINT TAB(12)"G TEILNEHMEN. DIE ANZAHL DER S
     PIELTAGE KANN MAXIMAL{3SPACE}H"                       <Q5G>
4770 PRINT TAB(12)"G 9999 TAGE BETRAGEN.{33SPACE}
     H"                                                    <PSG>
4780 PRINT TAB(12)"G DURCH GESCHICKTES KAUFEN UND
     VERKAUFEN MUSS VERSUCHTH"                             <BK5>
4790 PRINT TAB(12)"G WERDEN, DAS KAPITAL ZU VERME
     HREN.{19SPACE}H"                                      <77G>
4800 PRINT TAB(12)"G DIE KURSE AN DER BOERSE AEND
     ERN SICH TAEGLICH. VON{2SPACE}H"                      <EA9>
4810 PRINT TAB(12)"G{SHF.SPACE}ZEIT ZU ZEIT WERDE
     N NACHRICHTEN ANGEZEIGT, DIE{6SPACE}H"                <GQN>
4820 PRINT TAB(12)"G WICHTIGE HINWEISE AUF DIE KU
     RSENTWICKLUNG GEBEN.{4SPACE}H"                        <SB1>
4830 PRINT TAB(12)"G MOECHTEN SIE DAS SPIEL UNTER
     BRECHEN, SO KANN DER{4SPACE}H"                        <Q68>
4840 PRINT TAB(12)"G SPIELSTAND DURCH DRUECKEN DE
     R >F8< TASTE AUF DIS-{3SPACE}H"                       <044>
4850 PRINT TAB(12)"G KETTE GESPEICHERT WERDEN.{27
     SPACE}H"                                              <D00>
4860 PRINT TAB(12)"G" TAB(66)"H"                           <TF8>
4870 PRINT TAB(12)"G VIEL GLUECK BEIM SPEKULIEREN
      AN DER 'BOERSE 128'!{3SPACE}H"                       <IGC>
4880 PRINT TAB(12)"G" TAB(66)"H"                           <QF8>
4890 PRINT TAB(12)"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
     TTTTTTTTTTTTTTTTTTTTTTTTT"                            <3BM>
4900 GET KEY F$                                            <70A>
4910 EXIT                                                  <VJN>
4920 BEND                                                  <PRE>
4930 IF F$="N" THEN EXIT                                   <HBH>
4940 LOOP                                                  <B19>
4950 PRINT C6$+C7$+C1$+C2$+C7$                             <4NT>
4960 PRINT TAB(14)"MOECHTEN SIE EIN ALTES SPIEL W
     EITERSPIELEN ? [J/N]"                                 <UBD>
4970 DO                                                    <LEN>
4980 GET F$                                                <0SG>
4990 IF F$="J" THEN GOSUB 6150: RETURN                     <UQ2>
5000 IF F$="N" THEN GOSUB 5020: RETURN                     <JG3>
5010 LOOP                                                  <48P>
5020 REM                                                   <7KU>
5030 REM * DATEN FUER NEUES SPIEL *                        <7SV>
5040 REM                                                   <640>
5050 PRINT C6$+C7$+C1$+C2$+C7$                             <17T>
5060 PRINT TAB(17)"ANZAHL DER SPIELER [MAX 9 SPIE
     LER] :"                                               <MI6>
5070 PRINT                                                 <JIU>
5080 PRINT TAB(17)"ANZAHL SPIELTAGE [MAX 9999 TAG
     E]{3SPACE}:"                                          <FTH>
5090 IZ%=3: IS%=54: IL%=1: IN%=2: GOSUB 930: AS%=
     VAL(IP$)                                              <6F5>
5100 IZ%=5: IS%=54: IL%=4: IN%=1: GOSUB 930: AT%=
     VAL(IP$)                                              <607>
5110 DIM BE(15,AS%),SK(AS%),SA%(AS%,AF%),SN$(AS%)          <HHP>
5120 IZ%=7: IS%=54: IL%=15: IN%=0                          <LH7>
5130 FOR I1=1 TO AS%                                       <357>
5140 CHAR ,17,IZ%,"NAME SPIELER "+CHR$(I1+48)+"{2
     1SPACE}: "+C1$+C5$                                    <IG1>
5150 GOSUB 930: SN$(I1)=IP$                                <2C2>
5160 NEXT                                                  <I20>
5170 REM                                                   <6CP>
5180 REM * ANFANGSKURSE FESTLEGEN *                        <5KM>
5190 REM                                                   <5SN>
5200 FOR I=1 TO AF%                                        <09U>
5210 AK%(I)=100*(.95+.15*RND(0))                           <6TD>
5220 AE%(1,I)=AK%(I)                                       <Q7H>
5230 NEXT                                                  <U3G>
5240 FOR I=1 TO AS%                                        <D1B>
5250 SK(I)=25000                                           <F50>
5260 BE(1,I)=SK(I)                                         <F8P>
5270 NEXT                                                  <E1G>
5280 RETURN                                                <9C7>
5290 REM                                                   <4CH>
5300 REM * DATAZEILEN EINLESEN *                           <6KQ>
5310 REM                                                   <6SR>
```

```
5320 FOR I=1 TO AF%                                    <09U>
5330 READ NF$(I)                                       <1CF>
5340 AA%(I)=2000                                       <VAS>
5350 NEXT                                              <E1G>
5360 FOR I=1 TO 32                                     <4CU>
5370 READ FM$(I),FI%(I),MU%(I)                         <Q4L>
5380 NEXT                                              <Q70>
5390 KEY 1,C7$: KEY 3,C4$: KEY 5,CHR$(135): KEY 7
     ,CHR$(137): KEY 8,CHR$(138)                       <07I>
5400 RETURN                                            <947>
5410 REM                                               <5CL>
5420 REM * ABSCHLUSSBILANZ *                           <5KM>
5430 REM                                               <5SN>
5440 PRINT C6$+C7$+C1$+C2$                             <A0F>
5450 COLOR 5,4                                         <0QC>
5460 PRINT TAB(32)"ABSCHLUSSBILANZ"                    <JAS>
5470 FOR I1=1 TO AS%                                   <R17>
5480 FOR I=1 TO AF%                                    <89U>
5490 IF SA%(I1,I)>0 THEN BEGIN                         <E5F>
5500 SK(I1)=SK(I1)+SA%(I1,I)*AK%(I)*.98                <C95>
5510 SA%(I1,I)=0                                       <IK7>
5520 BEND                                              <9LE>
5530 NEXT                                              <64G>
5540 NEXT                                              <Q70>
5550 IF AS%>1 THEN BEGIN                               <JDB>
5560 FOR I1=1 TO AS%-1                                 <H58>
5570 FOR I=1 TO AS%-1                                  <H4Q>
5580 IF SK(I)<SK(I+1) THEN BEGIN                       <RQK>
5590 I2=SK(I): F$=SN$(I)                               <P2G>
5600 SK(I)=SK(I+1): SN$(I)=SN$(I+1)                    <GI0>
5610 SK(I+1)=I2: SN$(I+1)=F$                           <OLJ>
5620 BEND                                              <HME>
5630 NEXT                                              <U7G>
5640 NEXT                                              <240>
5650 BEND                                              <DLU>
5660 PRINT TAB(18)"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     @@@@@@@@@@@@@@"                                   <KUG>
5670 PRINT TAB(18)"G" TAB(61)"H"                       <FMH>
5680 FOR I=1 TO AS%                                    <51B>
5690 PRINT USING "{18SPACE}G{2SPACE}#.{3SPACE}###
     ############{7SPACE}#######{2SPACE}DM{2SPACE
     }H";I,SN$(I),SK(I)                                <BVR>
5700 PRINT TAB(18)"G" TAB(61)"H"                       <D6H>
5710 NEXT                                              <U7G>
5720 PRINT TAB(18)"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
     TTTTTTTTTTTTTT"                                   <69D>
5730 GET KEY F$                                        <10A>
5740 RETURN                                            <907>
5750 REM                                               <5SN>
5760 REM * SPIELSTAND SPEICHERN *                      <54K>
5770 REM                                               <5CL>
5780 PRINT C6$+C7$+C1$+C2$+C7$                         <1VT>
5790 PRINT TAB(19)"DER AKTUELLE SPIELSTAND WIRD G
     ESPEICHERT !"                                     <JLK>
5800 DO                                                <KEJ>
5810 DOPEN #1,"BOERSE.DAT",W                           <D5U>
5820 IF DS=0 THEN EXIT                                 <QQ3>
5830 IF DS=63 THEN DCLOSE : DOPEN #1,"@BOERSE.DAT
     ",W: EXIT                                         <SC2>
5840 DCLOSE                                            <8G7>
5850 CHAR ,19,5,"DISKETTENFEHLER - BITTE FEHLER A
     BSTELLEN ."                                       <DV9>
5860 GET KEY F$                                        <70A>
5870 LOOP                                              <S3P>
5880 PRINT#1,AS%                                       <BQF>
5890 PRINT#1,AT%                                       <B28>
5900 PRINT#1,TA%                                       <IRN>
5910 FOR I1=1 TO AS%                                   <BD6>
5920 PRINT#1,SK(I1)                                    <UCQ>
5930 FOR I=1 TO AF%                                    <ADU>
5940 PRINT#1,SA%(I1,I)                                 <K6E>
5950 NEXT                                              <UBG>
5960 PRINT#1,SN$(I1)                                   <E3V>
5970 NEXT                                              <MAG>
5980 FOR I=1 TO AF%                                    <SDU>
5990 PRINT#1,AK%(I)                                    <55M>
6000 PRINT#1,AA%(I)                                    <0B3>
6010 NEXT                                              <M6G>
6020 FOR I1=1 TO 15                                    <D2P>
6030 FOR I=1 TO AS%                                    <J1B>
6040 PRINT#1,BE(I1,I)                                  <BD4>
6050 NEXT                                              <64G>
6060 NEXT                                              <A50>
6070 FOR I1=1 TO 15                                    <72P>
6080 FOR I=1 TO AF%                                    <09U>
6090 PRINT#1,AE%(I1,I)                                 <VN8>
6100 NEXT                                              <Q70>
6110 NEXT                                              <U7G>
6120 DCLOSE                                            <907>
6130 PRINT C3$+CHR$(142)                               <97B>
6140 RETURN                                            <907>
6150 REM                                               <5SN>
6160 REM * SPIELSTAND LADEN *                          <54K>
6170 REM                                               <5CL>
6180 PRINT C6$+C7$+C1$+C2$+C7$                         <1VT>
6190 PRINT TAB(19)"DER GESPEICHERTE SPIELSTAND WI
     RD GELADEN !"                                     <Q52>
6200 DOPEN #1,"BOERSE.DAT"                             <N9U>
6210 IF DS>0 THEN BEGIN                                <DR0>
6220 DCLOSE                                            <9C7>
6230 PRINT                                             <JUU>
6240 PRINT TAB(19)"ES WURDE KEIN ALTER SPIELSTAND
      GEFUNDEN ."                                      <13N>
6250 SLEEP 3                                           <PA7>
6260 GOSUB 5020 : REM DATEN FUER NEUES SPIEL           <K73>
6270 RETURN                                            <PA7>
6280 BEND                                              <9HE>
6290 INPUT#1,AS%                                       <3SF>
6300 DIM BE(15,AS%),SK(AS%),SA%(AS%,AF%),SN$(AS%)      <3HP>
6310 INPUT#1,AT%                                       <M48>
6320 INPUT#1,TA%                                       <VTN>
6330 FOR I1=1 TO AS%                                   <317>
6340 INPUT#1,SK(I1)                                    <N8S>
6350 FOR I=1 TO AF%                                    <M9U>
6360 INPUT#1,SA%(I1,I)                                 <VFJ>
6370 NEXT                                              <M6G>
6380 INPUT#1,SN$(I1)                                   <TBM>
6390 NEXT                                              <E1G>
6400 FOR I=1 TO AF%                                    <09U>
6410 INPUT#1,AK%(I)                                    <C1G>
6420 INPUT#1,AA%(I)                                    <870>
6430 NEXT                                              <U3G>
6440 FOR I1=1 TO 15                                    <H2P>
6450 FOR I=1 TO AS%                                    <F1B>
6460 INPUT#1,BE(I1,I)                                  <APA>
6470 NEXT                                              <E1G>
6480 NEXT                                              <I60>
6490 FOR I1=1 TO 15                                    <B2P>
6500 FOR I=1 TO AF%                                    <49U>
6510 INPUT#1,AE%(I1,I)                                 <OV5>
6520 NEXT                                              <I20>
6530 NEXT                                              <M2G>
6540 DCLOSE                                            <9S7>
6560 RETURN                                            <9K7>
6570 REM                                               <7CT>
6580 REM * DATAS FIRMENNAMEN *                         <4KI>
6590 REM                                               <4SJ>
6600 DATA "APPLE","ACORN COMPUTER","ATARI","BROTH
     ER"                                               <7VO>
6610 DATA "CASIO","CITIZEN","COMMODORE","DIGITAL
     RESEARCH"                                         <IK1>
6620 DATA "EPSON","HEWLETT PACKARD","I.B.M.","NEC
     ","NIXDORF"                                       <KH1>
6630 DATA "OLIVETTI","ORIC","OSBORNE","PANASONOIC
     ","PHILIPS"                                       <UMP>
6640 DATA "SANYO","SCHNEIDER","SHARP","SINCLAIR",
     "SPECTRAVIDEO"                                    <KFN>
6650 DATA "TEXAS INSTRUMENTS"                          <7GL>
6660 REM                                               <6KQ>
6670 REM * DATAS FERNSCHREIBERMELDUNGEN *              <6SR>
6680 DATA "APPLE ENTWICKELT NEUEN SUPERCOMPUTER",
     1,15                                              <3OF>
6690 DATA "DIE FIRMA{2SPACE}ACRON COMPUTER{2SPACE
     }VERBUCHT UMSATZVERLUSTE",2,-10                   <G5F>
6700 DATA "NEUES MANAGEMENT BEI ATARI",3,-20           <6SH>
6710 DATA "PREISSENKUNGEN BEI BROTHER-DRUCKERN",4
     ,30                                               <2KD>
6720 DATA "DER NEUE TASCHENRECHNER VON CASIO KOMM
     T AUF DEN MARKT",5,20                             <UOM>
6730 DATA "CITIZEN KONNTE DEN ABSATZ IN DEUTSCHLA
     ND STEIGERN",6,25                                 <T0D>
6740 DATA "NEUER TRAUMCOMPUTER VON COMMODORE WURD
     E VORGESTELLT",7,30                               <8AC>
6750 DATA "DIGITAL RESEARCH ENTWICKELT NEUES BETR
     IEBSSYSTEM",8,25                                  <L3O>
6760 DATA "NEUER EPSON-DRUCKER WIRD SEHR GUT VERK
     AUFT",9,30                                        <QQ3>
6770 DATA "SCHLECHTE VERKAUFSBILANZEN BEI H.P.",1
     0,-20                                             <FK9>
6780 DATA "I.B.M. COMUTER WERDEN WIEDER MEHR VERK
     AUFT",11,10                                       <6E3>
6790 DATA "ROTE ZAHLEN BEI NEC",12,-20                 <A7T>
6800 DATA "SCHLECHTER ABSATZ BEI NIXDORF-COMPUTER
     N",13,-25                                         <05P>
6810 DATA "OLIVETTI SENKT PREISE FUER ALLE COMPUT
     ER",14,30                                         <IJ5>
6820 DATA "ORIC ENTWICKELT NEUEN COMPUTER FUER DE
     N HOME-BEREICH",15,15                             <FA6>
6830 DATA "NEUER OSBORNE-COMUTER KANN SICH NICHT
     DURCHSETZEN",16,-20                               <E4V>
6840 DATA "NEUE VERKAUFSPALETTE VON PANASONIC WUR
     DE VORGESTELLT",17,10                             <E0D>
6850 DATA "BILLIGER LASERDRUCKER VON PHILIPS KOMM
     T AUF DEN MARKT",18,20                            <07D>
6860 DATA "PREISE FUER SANYO-MONITORE WERDEN UM 1
     5% GESENKT",19,15                                 <3NE>
6870 DATA "VERKAUFSBILANZEN BEI SCHNEIDER WEITER
     GUT",20,20                                        <VOH>
6880 DATA "SHARP KOMMT MIT NEUEN HAND-HELD-COMPUT
     ER AUF DEN MARKT",21,15                           <61P>
6890 DATA "SINCLAIR-COMPUTER FINDEN KAUM NOCH ABN
     EHMER",22,-30                                     <1CL>
6900 DATA "NEUER HOME-COMPUTER VON SPECTRAVIDEO W
     IRD VORGESTELLT",23,10                            <EQB>
6910 DATA "T.I. STEIGT ERNEUT IN DEN HOME-COMPUTE
     RBERREICH EIN",24,20                              <OK9>
6920 DATA "ERNEUTE PREISSENKUNG BEI ATARI ST-COMP
     UTERN",3,15                                       <KBB>
6930 DATA "NIXDORF VERSTAERKT ENTWICKLUNGSARBEIT"
     ,13,10                                            <KC3>
6940 DATA "PRODUKTIONSSCHWIERIEGKEITEN BEI OLIVET
     TI-COMPUTERN",14,25                               <KKM>
6950 DATA "BESSERE VERKAUFSERFOLGE BEI OSBORNE",1
     6,10                                              <0Q2>
6960 DATA "CASIO VERBUCHT UMSATZVERLUSTE",5,-15        <96A>
6970 DATA "H.P. BRINGT NEUEN BILLIGCOMPUTER AUF D
     EN MARKT",10,10                                   <Q8H>
6980 DATA "COMMODORE MELDET LIEFERSCHWIRIEGKEITEN
      FUER AMIGA",7,-15                                <NBS>
6990 DATA "APPLE-COMPUTER WERDEN SCHLECHTER VERKA
     UFT",1,-20                                        <HFM>
```

Listing 1. (Schluß)

Vectors – Ein tolles Vergnügen



Ein Super-Action-Spiel für den C 128, das den Joystick heißlaufen läßt und Sie sicher für viele Stunden an den Bildschirm fesselt.

Dieses sehr schnelle Geschicklichkeitsspiel kann einen fast süchtig machen. Es greift die Idee zweier sich bekämpfender Schlangen auf (ähnlich der Motorrad-Sequenz im Film »Tron«) und setzt sie auf dem C128 im 80-Zeichen-Modus auf faszinierende Weise neu um. Bei »Vectors« kommt es darauf an, den Gegner durch geschicktes Manövrieren einzukreisen, ihn gegen eine Wand rennen zu lassen und damit auszuschalten. Durch die hohe Grafikauflösung (640 * 200 Punkte) ist der Kampf der beiden Linien wesentlich spannender als bei bisherigen Versionen. Wir hatten dieses Spiel zwar bereits in der Ausgabe 7/86 des 64'er-Magazins veröffentlicht, halten es aber für so gut (und es gibt ja auch kaum ein gutes Action-Spiel für den C128), daß wir unseren neuen Lesern diesen Spaß nicht vorenthalten wollen.

Vectors wurde mit einigen Zusatzfunktionen ausgestattet. So kann man als weitere Schwierigkeiten zufällige Hindernisse in das Spielfeld einschalten, die auch vom Benutzer editierbar sind. Auch diagonale Linien sind möglich, wodurch die Schwierigkeit für Vector-Profis gesteigert werden kann.

Es besteht aber auch die Möglichkeit, gegen den Computer anzutreten und verschiedene Schwierigkeiten anzuwählen. Diese können durch Ab- und Hinzuschalten verschiedener Computerstrategien zusätzlich variiert werden.

Hat man den Menüpunkt »Beschleunigen« gewählt, so läßt sich durch Drücken des Feuerknopfes der Vector (Linie) beschleunigen.

Computerstrategien

Die Spielstärke des Computers wird durch Hinzuschalten verschiedener Strategien eingestellt. Im Programm sind die zehn sinnvollsten Kombinationen der folgenden Einzelstrategien integriert:

- Spirale
- Gasse
- Folgen
- Besoffen
- Zufall

Der Computer-Vector ist prinzipiell so gesteuert, daß er zwei Pixel vor einem Hindernis (Gegner-Vector, Hindernis oder Spielfeldrand) ausweicht. Durch dieses Freilassen eines Pixels sichert er sich einen Ausweg, falls er in eine Falle läuft.

Beim Ausweichen wird die neue Richtung zufällig ausgewählt (siehe auch »Folgen«).

Die Strategien im einzelnen

Spirale: Diese Strategie überprüft, ob sich der Computer-Vector in einer Spirale verfangen hat und sichert sein Entkommen.

Gasse: Steuert der Computer auf einen Spalt zu, der nur einen Punkt breit ist (mögliche Sackgasse), so weicht er aus.

Folgen: Bei einem Ausweichmanöver orientiert sich der Computer an der aktuellen Position des Gegner-Vectors. Man wird von ihm verfolgt!

Besoffen: Der Computer weicht nach einer bestimmten Anzahl zurückgelegter Pixel aus (siehe auch »Folgen«), egal, ob er auf ein Hindernis stößt oder nicht. Damit wird er unberechenbar!

Zufall: Diese Strategie arbeitet mit »Besoffen« zusammen. Sie bestimmt die Anzahl der Pixel, nach der der Computer die Richtung wechselt. Falls die Strategie aktiviert ist, addiert er zu einem Startwert einen Zufallswert, ansonsten übernimmt er den Startwert.

Die Spielstärke ergibt sich nicht unmittelbar aus der Reihenfolge der Strategiekombinationen. Unserer Meinung nach ist die Strategie 7 die spielstärkste. Sehenswert ist aber auch Strategie 4.

Ändern der Hindernisse

Nach dem Laden des Programms befinden sich im Speicher zwei Tabellen. Die erste Tabelle liegt ab der Speicherstelle $07374 in BANK 0. Es folgen 16 Byte, die jeweils den Wert 0 oder 1 haben. Der Wert 1 steht für »Hindernis gesetzt«, 0 bedeutet »Hindernis aus«.

Dabei entspricht Byte 0=Hindernis 0, Byte 2=Hindernis 2 und so weiter. Hieraus ergibt sich eine maximale Anzahl von 16 Hindernissen.

In Tabelle 2 (ab Adresse $07384, BANK 0) sind für jedes Hindernis zwei Byte reserviert, die den Anfang eines Hindernisses in Low-Byte und High-Byte angeben:

Byte 0 und 1: Low-Byte und High-Byte der Anfangsadresse des Hindernisses Nummer 0;

Byte 2 und 3: Low-Byte und High-Byte der Anfangsadresse des Hindernisses Nummer 1;

und so weiter...

Die Hindernisse können ab der Adresse $07643 in BANK 0 abgelegt werden.

Aufbau eines Hindernisses

Ein Hindernis baut sich aus vertikalen, horizontalen und diagonalen Strichen auf. Ein Strich setzt sich aus fünf Byte zusammen:

- Byte 0: Low-Byte der X-Koordinate der Startkoordinaten des Striches
- Byte 1: High-Byte der X-Koordinate der Startkoordinaten des Striches
- Byte 2: Y-Koordinate der Startkoordinaten des Striches
- Byte 3: Länge des Striches (maximal 255)
- Byte 4: Richtung des Striches
  - 0 - Rechts
  - 1 - Unten
  - 2 - Links/Unten
  - 3 - Rechts/Unten

Ein Strich mit den Koordinaten X=317, Y=97 und der Länge 46 in Richtung Rechts/Unten sähe so aus:

Byte:	0	1	2	3	4
Wert:	061	001	097	046	003

Zum Austesten empfiehlt es sich, den eingebauten Monitor zu benutzen.

Eingabe des Programms

Das Hauptprogramm (Listing 1) geben Sie mit dem MSE in C 64-Modus ein und speichern es auf Ihren Datenträger. Danach schalten Sie den Computer bitte aus und ein, um in den C 128-Modus zu gelangen.

Hier geben Sie noch die nachstehende Zeile 10 ein und speichern diese unter dem Namen VECTORS.BOOT auf den gleichen Datenträger (dies ist das spätere Ladeprogramm)

```
10 BLOAD"VECTORS.OBJ",ON B0:BANK 0:SYS 24576
```

Die Bewegung der Vectors und die Auswahl der Menüpunkte erfolgt mit dem Joystick (80-Zeichenschirm einschalten).

Wem die Hindernisse und Strategien trotzdem noch zu leicht sind, der sollte ruhig das Spiel erweitern. So zum Beispiel einen dritten, unabhängigen Vector, der jeden der beiden anderen behindert. Oder knobeln Sie sich selbst noch weitere, ausgekochte Hindernisse aus.

(H. Beine/A. Jansen/kn)

```
name : vectors.obj       6000 7642
-----------------------------------
6000 : ad 00 ff 8d 13 5f a2 ff  16
6008 : 9a a9 0e 8d 00 ff a2 19  69
6010 : 20 c2 62 29 07 8d 30 5f  ab
6018 : a9 fc cd 12 d0 90 fb a9  ca
6020 : 00 8d 20 d0 8d 21 d0 a9  81
6028 : 00 8d 11 d0 a9 01 8d 30  86
6030 : d0 85 28 85 68 a9 00 8d  6d
6038 : 02 dc 8d 03 dc 20 81 62  06
6040 : 20 b7 62 4c ac 69 85 69  5d
6048 : 86 6a 98 29 7f 8d 07 5f  8e
6050 : 98 29 80 8d 08 5f 20 d5  f6
6058 : 72 20 6b 66 a5 69 85 27  8c
6060 : a9 01 85 aa a9 00 85 ab  48
6068 : 8d 31 5f 8d 32 5f a9 32  41
6070 : 85 a9 a9 00 8d 00 5f 8d  a6
6078 : 01 5f 8d 02 5f a9 00 a2  55
6080 : 00 a0 00 20 dd 62 a0 c7  d7
6088 : 20 dd 62 18 69 01 90 01  15
6090 : e8 e0 02 d0 ec c9 80 d0  44
6098 : e8 a9 02 85 6b 85 6c a0  5c
60a0 : 00 a9 00 a2 00 20 dd 62  06
60a8 : a9 7f a2 02 20 dd 62 c8  06
60b0 : c0 c8 d0 ed a5 6a 29 20  59
60b8 : f0 03 20 53 73 a9 32 a2  2f
60c0 : 00 a0 64 85 23 86 24 84  da
60c8 : 25 a9 f7 85 26 a9 4e a2  9f
60d0 : 02 a0 64 85 63 86 64 84  f1
60d8 : 65 a9 fb 85 66 a5 23 a6  2f
60e0 : 24 a4 25 20 dd 62 a6 26  7c
60e8 : a0 00 20 18 66 24 6a 10  e5
60f0 : 13 aa 29 10 08 8a 28 f0  fc
60f8 : 0b c6 6b f0 03 4c 9f 61  33
6100 : a2 03 86 6b 24 6a 50 02  0e
6108 : 09 10 aa 18 65 26 c9 fb  6e
6110 : f0 15 c9 f2 f0 11 8a c9  b1
6118 : fe f0 0e c9 fd f0 0a c9  6e
6120 : fb f0 06 c9 f7 f0 02 a5  a8
6128 : 26 85 26 8a 29 10 45 26  60
6130 : 8d 26 5f a5 26 c9 fe d0  ab
6138 : 0d c6 28 d0 62 a9 02 85  53
6140 : 28 c6 25 4c 72 61 c9 fd  f4
6148 : d0 0d c6 28 d0 51 a9 02  98
6150 : 85 28 e6 25 4c 72 61 c9  b9
6158 : fb d0 0d c6 23 a5 23 c9  57
6160 : ff d0 02 c6 24 4c 72 61  52
6168 : c9 f7 d0 06 e6 23 d0 02  f0
6170 : e6 24 a5 24 c9 02 d0 09  58
6178 : a5 23 c9 80 d0 03 4c cb  1f
6180 : 67 a5 25 c9 ff d0 03 4c  67
6188 : cb 67 c9 c8 d0 03 4c cb  80
6190 : 67 a5 23 a6 24 a4 25 20  a4
6198 : da 62 90 03 4c cb 67 a6  36
61a0 : 66 a0 01 20 18 66 aa a5  45
61a8 : 6a 29 10 f0 07 20 78 63  e3
61b0 : 8d 32 5f aa 8a 24 6a 10  17
61b8 : 13 aa 29 10 08 8a 28 f0  c4
61c0 : 0b c6 6c f0 03 4c 6b 62  6d
61c8 : a2 03 86 6c 24 6a 50 02  f6
61d0 : 09 10 aa 18 65 66 c9 fb  38
61d8 : f0 19 8a 18 65 66 c9 f2  91
61e0 : f0 11 8a c9 fe f0 0e c9  78
61e8 : fd f0 0a c9 fb f0 06 c9  0c
61f0 : f7 f0 02 a5 66 85 66 8a  d6
61f8 : 29 10 45 66 8d 27 5f a5  22
6200 : 66 c9 fe d0 0d c6 68 d0  6f
6208 : 62 a9 02 85 68 c6 65 4c  5b
6210 : 3e 62 c9 fd d0 0d c6 68  13
6218 : d0 51 a9 02 85 68 e6 65  3d
6220 : 4c 3e 62 c9 fb d0 0d c6  65
6228 : 63 a5 63 c9 ff d0 02 c6  8c
6230 : 64 4c 3e 62 c9 f7 d0 06  42
6238 : e6 63 d0 02 e6 64 a5 64  35
6240 : c9 02 d0 09 a5 63 c9 80  fd
6248 : d0 03 4c d3 67 a5 65 c9  f4
6250 : ff d0 03 4c d3 67 c9 c8  33
6258 : d0 03 4c d3 67 a5 63 a6  b6
6260 : 64 a4 65 20 da 62 90 03  7d
6268 : 4c d3 67 a5 63 a6 64 a4  73
6270 : 65 20 dd 62 a4 27 a2 01  b9
6278 : ca d0 fd 88 d0 fa 4c dd  0c
6280 : 60 a0 64 b9 8f 62 99 00  f3
6288 : 01 88 10 f7 4c 00 01 78  8a
6290 : a9 7f 8d 00 ff a9 17 8d  21
6298 : f8 ff a9 01 8d f9 ff a9  16
62a0 : 0e 8d 00 ff 58 60 20 09  90
62a8 : e1 20 56 e0 20 00 c0 a9  a3
62b0 : 0e 8d 00 ff 4c 00 60 a9  1e
62b8 : 17 8d 18 03 a9 01 8d 19  07
62c0 : 03 60 8e 00 d6 2c 00 d6  13
62c8 : 10 fb ad 01 d6 60 8e 00  0c
62d0 : d6 2c 00 d6 10 fb 8d 01  b0
62d8 : d6 60 18 90 01 38 85 1b  15
62e0 : 86 1c 84 1d 08 85 1e 86  6b
62e8 : 1f 84 20 a5 1e 85 22 46  29
62f0 : 1f 66 1e 46 1f 66 1e 46  bd
62f8 : 1f 66 1e a9 00 85 21 a5  03
6300 : 20 06 20 26 21 06 20 26  ff
6308 : 21 65 20 85 20 90 02 e6  f1
6310 : 21 a2 04 06 20 26 21 ca  92
6318 : d0 f9 a5 1e 65 20 85 20  c0
6320 : 90 02 e6 21 a2 12 a5 21  23
6328 : 20 ce 62 e8 a5 20 20 ce  de
6330 : 62 a2 1f 20 c2 62 48 a5  5b
6338 : 22 29 07 aa 68 28 90 2b  66
6340 : 1d 63 63 48 a2 12 a5 21  84
6348 : 20 ce 62 e8 a5 20 20 ce  fe
6350 : 62 a2 1f 68 20 ce 62 a2  1f
6358 : 12 20 ce 62 a5 1b a6 1c  80
6360 : a4 1d 60 80 40 20 10 08  10
6368 : 04 02 01 3d 63 63 f0 01  6c
6370 : 38 a5 1b a6 1c a4 1d 60  33
6378 : a2 00 a0 00 ad 00 5f 10  bb
6380 : 04 8e 00 5f 60 a5 66 c9  18
6388 : fe f0 04 c9 fd d0 06 a5  02
6390 : 68 c9 02 d0 03 ee 02 5f  e6
6398 : a5 66 c9 fe d0 04 a2 00  7a
63a0 : a0 ff c9 fd d0 04 a2 00  2a
63a8 : a0 01 c9 fb d0 04 a2 ff  72
63b0 : a0 00 c9 f7 d0 04 a2 01  7b
63b8 : a0 00 86 a7 84 a8 a5 63  da
63c0 : 85 a4 a5 64 85 a5 a5 65  74
63c8 : 85 a6 a5 a7 f0 1c 10 0d  49
63d0 : c6 a4 a5 a4 c9 ff d0 12  ea
63d8 : c6 a5 4c ea 63 18 a5 a4  b8
63e0 : 65 a7 85 a4 a5 a5 69 00  3c
63e8 : 85 a5 a5 a6 18 65 a8 85  d8
63f0 : a6 a5 a4 a6 a5 a4 a6 20  c1
63f8 : da 62 90 07 a9 32 85 a9  9e
6400 : 4c 63 64 a5 a7 f0 1c 10  5e
6408 : 0d c6 a4 a5 a4 c9 ff d0  90
6410 : 12 c6 a5 4c 23 64 18 a5  79
6418 : a4 65 a7 85 a4 a5 a5 69  ea
6420 : 00 85 a5 a5 a6 18 65 a8  13
```

Listing 1. »Vectors« – ein schnelles Action-Spiel für den C128. Bitte mit dem MSE (Seite 158) im C64-Modus eingeben

```
6428 : 85 a6 a5 a4 a6 a5 a4 a6   76
6430 : 20 da 62 90 07 a9 32 85   fa
6438 : a9 4c 63 64 20 c5 65 a5   7e
6440 : 6a 29 08 f0 04 c6 a9 f0   5e
6448 : 05 a9 ff 4c 3a 65 a5 6a   e5
6450 : 29 02 08 a9 00 28 f0 05   c1
6458 : ad 12 d0 29 4f 18 6d 07   e1
6460 : 5f 85 a9 ad 02 5f c9 02   e8
6468 : d0 03 4c 5c 65 a9 04 85   17
6470 : a8 20 8e 65 85 a7 a5 63   6c
6478 : 85 a4 a9 00 8d 02 5f a5   6b
6480 : 64 85 a5 a5 65 85 a6 20   22
6488 : 06 65 b0 21 20 06 65 b0   ba
6490 : 1c a6 a7 a9 00 38 2a ca   1e
6498 : 10 fc c5 66 f0 0f 49 ff   11
64a0 : 20 3a 65 ae 03 5f 8e 01   74
64a8 : 5f 8d 03 5f 60 e6 a7 a5   a2
64b0 : a7 c9 04 d0 02 a9 00 85   cf
64b8 : a7 c6 a8 d0 b9 a9 04 85   0a
64c0 : a8 a5 63 85 a4 a5 64 85   d8
64c8 : a5 a5 65 85 a6 20 06 65   98
64d0 : b0 1c a6 a7 a9 00 38 2a   fd
64d8 : ca 10 fc c5 66 f0 0f 49   5f
64e0 : ff 20 3a 65 ae 03 5f 8e   c8
64e8 : 01 5f 8d 03 5f 60 e6 a7   40
64f0 : a5 a7 c9 04 d0 02 a9 00   1f
64f8 : 85 a7 c6 a8 d0 c3 a5 66   a6
6500 : 8d 01 5f a9 ff 60 a5 a7   04
6508 : c9 00 d0 05 c6 a6 4c 31   db
6510 : 65 c9 01 d0 05 e6 a6 4c   6f
6518 : 31 65 c9 02 d0 0d c6 a5   88
6520 : a5 a4 c9 ff d0 02 c6 a5   0d
6528 : 4c 31 65 e6 a4 d0 02 e6   ea
6530 : a5 a5 a4 a6 a5 a4 a6 4c   58
6538 : da 62 85 10 24 6a 50 19   b0
6540 : c6 aa d0 0b ad 12 d0 85   ab
6548 : aa a5 ab 49 ff 85 ab a5   fc
6550 : ab 30 06 a5 10 49 10 85   e0
6558 : 10 a5 10 60 a5 6a 29 01   9f
6560 : f0 2b a9 00 8d 02 5f a5   02
6568 : 66 49 ff 29 03 08 a5 66   6c
6570 : 28 f0 05 49 03 4c 7a 65   c2
6578 : 49 0c 49 ff 20 3a 65 8d   9e
6580 : 00 5f a9 02 8d 28 5f ad   cd
6588 : 01 5f 4c 3a 65 60 a5 6a   58
6590 : 29 04 08 ad 12 d0 29 03   c5
6598 : 28 f0 29 a5 66 c9 fb f0   be
65a0 : 04 c9 f7 d0 0e a5 25 c5   cf
65a8 : 65 90 04 a9 01 d0 15 a9   ca
65b0 : 00 f0 11 a5 23 38 e5 63   74
65b8 : a5 24 e5 64 90 04 a9 03   4b
65c0 : d0 02 a9 02 60 ad 08 5f   8e
65c8 : 10 4d a5 a7 f0 22 a5 a4   dd
65d0 : a6 a5 a4 a6 88 20 da 62   00
65d8 : 90 3d a5 a4 a6 a5 a4 a6   7c
65e0 : c8 20 da 62 90 31 a9 01   f6
65e8 : 8d 28 5f 68 68 4c 63 64   ad
65f0 : a6 a5 a5 a4 18 69 01 90   59
65f8 : 01 e8 a4 a6 20 da 62 90   ef
6600 : 16 a6 a5 a5 a4 38 e9 01   3d
6608 : b0 01 ca a4 a6 20 da 62   1c
6610 : 90 05 68 68 4c 63 64 60   7c
6618 : b9 00 dc 09 10 c9 ff d0   1b
6620 : 03 b9 31 5f 29 ef 99 31   13
6628 : 5f b9 00 dc 29 10 19 31   d9
6630 : 5f 99 31 5f ad 39 5f f0   98
6638 : 2e 8a 09 10 49 ff 99 33   51
6640 : 5f b9 31 5f 09 10 49 ff   ea
6648 : 39 33 5f f0 1a b9 31 5f   04
6650 : 19 33 5f 29 ef 99 35 5f   5f
6658 : b9 31 5f 29 10 19 35 5f   04
6660 : 60 b9 31 5f 09 ef 60 b9   da
6668 : 31 5f 60 78 a9 00 8d 14   69
6670 : 03 a9 02 8d 15 03 a2 08   7e
6678 : bd fd 66 9d 00 02 ca 10   dc
6680 : f7 a9 00 8d 26 5f 8d 27   df
6688 : 5f 8d 28 5f a9 00 8d 1a   a9
6690 : d0 8d 19 d0 ad 04 0a 29   fd
6698 : fe 8d 04 0a a2 18 1e 00   03
66a0 : d4 ca 10 fa a9 f0 8d 0d   af
66a8 : d4 a9 b0 8d 06 d4 a9 f0   be
66b0 : 8d 14 d4 a9 15 8d 04 d4   29
66b8 : a9 11 8d 0b d4 a9 9f 8d   e3
66c0 : 18 d4 a9 00 8d 0f dc a9   c5
66c8 : 7f 8d 0d dc a9 81 8d 0d   e4
66d0 : dc a9 0e 8d 05 dc a9 01   96
66d8 : 8d 0e dc a9 00 8d 2c 5f   b4
66e0 : 8d 2d 5f 8d 2e 5f 8d 29   f4
66e8 : 5f 8d 2a 5f a9 1e 8d 01   48
66f0 : d4 a9 f2 8d 17 d4 a9 01   c8
66f8 : 8d 2f 5f 58 60 a9 0e 8d   a6
6700 : 00 ff 4c 05 67 d8 ad 0d   c2
6708 : dc ad 31 5f 29 ef cd 29   8f
6710 : 5f f0 08 8d 29 5f a9 00   cf
6718 : 8d 2c 5f ad 32 5f 29 ef   eb
6720 : cd 2a 5f f0 08 8d 2a 5f   4d
6728 : a9 00 8d 2e 5f ce 2f 5f   e2
```

```
6730 : d0 0e a9 0a 8d 2f 5f ee   61
6738 : 2c 5f ad 2c 5f 8d 0f d4   4d
6740 : ee 2d 5f ad 2d 5f 10 0e   7d
6748 : ad 2e 5f 18 69 05 8d 2e   39
6750 : 5f a9 00 8d 2d 5f ad 2e   16
6758 : 5f 18 69 0f 2d 2d 5f 8d   d4
6760 : 08 d4 4c 33 ff 78 a9 08   c6
6768 : 8d 04 d4 8d 0b d4 8d 12   90
6770 : d4 a9 00 8d 0e dc 8d 0f   e7
6778 : dc a9 65 a2 fa 8d 14 03   49
6780 : 8e 15 03 ad 0d dc 58 60   e9
6788 : 20 65 67 a2 18 1e 00 d4   a5
6790 : ca 10 fa a9 03 8d 0f d4   d9
6798 : a9 fc 8d 14 d4 a9 f4 8d   2f
67a0 : 17 d4 a9 1f 8d 18 d4 a9   b0
67a8 : 81 8d 12 d4 a9 80 8d 12   08
67b0 : d4 a9 0c a2 00 a0 00 ca   4b
67b8 : d0 fd 88 d0 fa 38 e9 01   de
67c0 : d0 f5 a9 00 8d 17 d4 8d   f5
67c8 : 12 d4 60 a2 01 ee 24 5f   88
67d0 : 4c d8 67 a2 02 ee 25 5f   a1
67d8 : 86 67 20 88 67 20 fa 72   73
67e0 : 20 d5 72 a9 3e a2 69 20   9b
67e8 : d1 70 a5 67 ad 15 5f 29   9b
67f0 : 10 f0 1a a5 67 c9 01 d0   1e
67f8 : 0a a9 be a2 68 20 d1 70   8a
6800 : 4c 24 68 a9 cc a2 68 20   71
6808 : d1 70 4c 24 68 a5 67 c9   8e
6810 : 01 d0 0a a9 d9 a2 68 20   c6
6818 : d1 70 4c 24 68 a9 e3 a2   62
6820 : 68 20 d1 70 ce 23 5f f0   80
6828 : 17 a9 ed a2 68 20 d1 70   93
6830 : ad 23 5f a2 30 a0 0e 20   1b
6838 : 35 72 20 80 69 4c 56 60   d2
6840 : 20 d5 72 a9 0a a2 69 20   b8
6848 : d1 70 a9 3e a2 69 20 d1   1d
6850 : 70 a9 36 a2 69 20 d1 70   36
6858 : ad 15 5f 29 10 f0 07 a9   85
6860 : be a2 68 4c 6a 68 a9 d9   57
6868 : a2 68 20 d1 70 a2 28 a0   7f
6870 : 0e ad 24 5f 20 35 72 20   00
6878 : b7 68 a9 3a a2 69 20 d1   af
6880 : 70 ad 15 5f 29 10 f0 07   dd
6888 : a9 cc a2 68 4c 93 68 a9   a3
6890 : e3 a2 68 20 d1 70 a2 28   5e
6898 : a0 10 ad 25 5f 20 35 72   01
68a0 : 20 b7 68 20 80 69 20 [illegible]5   39
68a8 : 72 a9 45 a2 73 20 d1 70   f5
68b0 : 20 80 69 20 65 67 60 a9   d5
68b8 : 4c a2 69 4c d1 70 44 45   75
68c0 : 52 20 43 4f 4d 50 55 54   32
68c8 : 45 52 20 00 44 45 52 20   36
68d0 : 53 50 49 45 4c 45 52 20   bf
68d8 : 00 53 50 49 45 4c 45 52   30
68e0 : 20 31 00 53 50 49 45 4c   00
68e8 : 45 52 20 32 00 01 17 0e   25
68f0 : 4e 4f 43 48 20 5a 55 20   2a
68f8 : 53 50 49 45 4c 45 4e 44   1f
6900 : 45 20 52 55 4e 44 45 4e   4d
6908 : 3a 00 01 00 00 02 2a 2a   90
6910 : 2a 20 45 4e 44 45 52 47   ac
6918 : 45 42 4e 49 53 20 2a 2a   6e
6920 : 2a 20 2d 2d 2d 2d 2d 2d   97
6928 : 2d 2d 2d 2d 2d 2d 2d 2d   28
6930 : 2d 2d 2d 2d 2d 00 01 1b   f2
6938 : 0e 00 01 1b 10 00 01 1b   25
6940 : 0c 45 53 20 53 49 45 47   eb
6948 : 54 45 20 00 20 20 20 20   0b
6950 : 20 20 4d 41 4c 00 64 00   52
6958 : 19 96 01 1b 02 19 96 01   a5
6960 : c8 00 1e ef 00 c8 00 aa   49
6968 : ef 00 22 01 55 1e 01 22   8f
6970 : 01 55 3c 00 5e 01 55 1e   aa
6978 : 01 22 01 73 3c 00 ff ff   fd
6980 : 20 65 67 ad 00 dc 2d 01   80
6988 : dc 29 10 f0 f6 ad 00 dc   b2
6990 : 2d 01 dc 29 10 d0 f6 20   3e
6998 : 8e c9 a2 00 a0 00 ca d0   8a
69a0 : fd 88 d0 fa 60 56 45 43   c9
69a8 : 54 4f 52 53 a0 06 b9 00   c4
69b0 : 20 d9 a5 69 d0 06 88 10   d3
69b8 : f5 4c f8 69 a0 06 b9 a5   ab
69c0 : 69 99 00 20 88 10 f7 24   2b
69c8 : d7 30 0a a9 1b 20 d2 ff   6d
69d0 : a9 58 20 d2 ff a9 93 20   e4
69d8 : d2 ff a9 05 8d 14 5f a9   ff
69e0 : 40 8d 15 5f a9 00 8d 16   15
69e8 : 5f a9 01 8d 17 5f a9 0a   35
69f0 : 8d 18 5f a9 00 8d 39 5f   a6
69f8 : a9 05 8d 1b 5f 8d 1c 5f   7c
6a00 : 20 d5 72 a9 27 a2 6b 20   52
6a08 : d1 70 a9 7c a2 6d 20 40   a2
6a10 : 70 a9 8f a2 6b 20 d1 70   6d
6a18 : a9 5c a2 6c 20 d1 70 ad   d3
6a20 : 15 5f 29 10 d0 0a a9 3d   b0
6a28 : a2 6b 20 d1 70 4c 37 6a   dd
6a30 : a9 4a a2 6b 20 d1 70 a2   ac
```

```
6a38 : 43 a0 10 ad 14 5f 20 35   ac
6a40 : 72 a2 13 a0 12 ad 15 5f   7e
6a48 : 29 80 20 8b 72 a2 3f a0   a5
6a50 : 12 ad 15 5f 29 20 20 8b   95
6a58 : 72 a2 13 a0 14 ad 18 5f   c2
6a60 : 20 35 72 ad 15 5f 29 10   7e
6a68 : d0 0a a9 6a a2 6b 20 d1   9f
6a70 : 70 4c 85 6a a9 57 a2 6b   6c
6a78 : 20 d1 70 a2 3f a0 16 ad   9e
6a80 : 17 5f 20 35 72 a0 16 a2   bf
6a88 : 0d ad 39 5f 20 8b 72 ad   2a
6a90 : 00 dc cd 00 dc d0 f8 c9   3d
6a98 : ff f0 f4 48 20 8e c9 68   c4
6aa0 : ac 1b 5f 8c 1c 5f c9 fe   25
6aa8 : d0 0b ce 1b 5f 10 03 ee   75
6ab0 : 1b 5f 4c d4 6a c9 fd f0   f7
6ab8 : 06 20 61 6c 4c 1f 6a ee   f9
6ac0 : 1b 5f a2 07 ad 15 5f 29   68
6ac8 : 10 f0 01 e8 ec 1b 5f d0   74
6ad0 : 03 ce 1b 5f ad 1c 5f 20   67
6ad8 : c6 72 8e c1 5f 8c c2 5f   d7
6ae0 : a9 01 8d c0 5f a9 20 8d   64
6ae8 : c3 5f a9 00 8d c4 5f a9   95
6af0 : c0 a2 5f 20 d1 70 ad 1b   6b
6af8 : 5f 20 c6 72 8e c1 5f 8c   f5
6b00 : c2 5f a9 3e 8d c3 5f a9   6c
6b08 : c0 a2 5f 20 d1 70 a2 00   20
6b10 : a0 00 88 d0 fd ca d0 fa   5c
6b18 : ad 00 dc cd 00 dc d0 f8   d2
6b20 : c9 ff d0 f4 4c 8f 6a 01   a8
6b28 : 20 00 02 4f 52 42 49 54   b8
6b30 : 02 01 26 03 d0 d2 c5 d3   ff
6b38 : c5 ce d4 d3 00 01 0c 10   6c
6b40 : 02 53 50 49 45 4c 45 52   9a
6b48 : 20 00 01 0c 10 02 43 4f   e7
6b50 : 4d 50 55 54 45 52 00 01   8e
6b58 : 2f 16 38 29 20 43 4f 4d   ba
6b60 : 50 55 54 45 52 20 49 51   07
6b68 : 3a 00 01 2f 16 20 20 20   ec
6b70 : 20 20 20 20 20 20 20 20   70
6b78 : 20 20 20 20 20 20 20 20   78
6b80 : 20 20 20 20 20 20 00 4a   54
6b88 : 41 20 20 4e 45 49 4e 01   85
6b90 : 01 10 31 29 20 47 45 47   eb
6b98 : 4e 45 52 3a 01 2f 10 32   93
6ba0 : 29 20 47 45 53 43 48 57   73
6ba8 : 49 4e 44 49 47 4b 45 49   c9
6bb0 : 54 3a 01 01 12 33 29 20   21
6bb8 : 42 45 53 43 48 4c 45 55   81
6bc0 : 4e 49 47 45 4e 3a 01 2f   46
6bc8 : 12 34 29 20 48 49 4e 44   d3
6bd0 : 45 52 4e 49 53 53 45 3a   54
6bd8 : 01 01 14 35 29 20 41 4e   3b
6be0 : 5a 41 48 4c 20 52 55 4e   fd
6be8 : 44 45 4e 3a 01 2f 14 36   f0
6bf0 : 29 20 02 53 54 41 52 54   56
6bf8 : 02 01 01 16 37 29 20 44   44
6c00 : 49 41 47 4f 4e 41 4c 3a   3a
6c08 : 01 00 18 50 52 4f 47 52   7b
6c10 : 41 4d 20 57 52 49 54 54   54
6c18 : 45 4e 20 49 4e 20 31 39   d3
6c20 : 38 36 20 42 59 20 41 52   04
6c28 : 4e 45 20 4a 41 4e 53 45   c9
6c30 : 4e 20 26 20 48 41 52 41   76
6c38 : 4c 44 20 42 45 49 4e 45   59
6c40 : 20 20 2d 20 20 28 43 29   62
6c48 : 20 43 4f 50 59 52 49 47   c4
6c50 : 48 54 20 42 59 20 36 34   eb
6c58 : 27 45 52 00 01 2e 14 3e   05
6c60 : 00 8d 20 5f ad 1b 5f 0a   60
6c68 : aa bd 78 6c 8d 21 5f bd   77
6c70 : 79 6c 8d 22 5f 6c 21 5f   64
6c78 : 88 6c 91 6c b3 6c bc 6c   93
6c80 : c5 6c 43 6d e7 6c f0 6c   78
6c88 : ad 15 5f 49 10 8d 15 5f   41
6c90 : 60 ad 20 5f c9 fb d0 09   8d
6c98 : ce 14 5f d0 15 ee 14 5f   3a
6ca0 : 60 c9 f7 d0 0d ee 14 5f   54
6ca8 : ad 14 5f c9 10 d0 03 ce   a2
6cb0 : 14 5f 60 ad 15 5f 49 80   b4
6cb8 : 8d 15 5f 60 ad 15 5f 49   47
6cc0 : 20 8d 15 5f 60 ad 20 5f   8b
6cc8 : c9 fb d0 09 ce 18 5f d0   b1
6cd0 : e1 ee 18 5f 60 c9 f7 d0   f0
6cd8 : d9 ee 18 5f ad 18 5f c9   c7
6ce0 : 64 d0 cf ce 18 5f 60 ad   d3
6ce8 : 39 5f 49 01 8d 39 5f 60   24
6cf0 : ad 20 5f c9 fb d0 09 ce   c6
6cf8 : 17 5f d0 15 ee 17 5f 60   7b
6d00 : c9 f7 d0 0d ee 17 5f ad   1b
6d08 : 17 5f c9 0b d0 03 ce 17   31
6d10 : 5f ae 17 5f ca bd 39 6d   d2
6d18 : 29 0f 8d 21 5f bd 39 6d   f4
6d20 : 29 f0 8d 22 5f ad 15 5f   df
6d28 : 29 f0 0d 21 5f 8d 15 5f   a6
6d30 : ad 22 5f 09 01 8d 16 5f   7b
6d38 : 60 00 01 05 8d 8f af ff   8d
```

```
6d40 : 8a 81 0a ad 20 5f c9 ef  c7
6d48 : f0 01 60 ad 18 5f 8d 23  80
6d50 : 5f a9 00 8d 24 5f 8d 25  f3
6d58 : 5f ae 14 5f bd 6c 6d ae  52
6d60 : 15 5f ac 16 5f 20 46 60  e4
6d68 : 68 68 4c 00 6a 01 02 04  d6
6d70 : 07 0a 0f 14 28 50 64 82  5e
6d78 : a0 c8 e6 00 00 00 00 64  ff
6d80 : 03 c8 00 00 64 02 00 00  3e
6d88 : 00 05 00 05 00 00 5f 03  2f
6d90 : c3 00 00 5f 02 be 00 00  55
6d98 : 05 00 be 00 00 5a 02 0a  3c
6da0 : 00 00 5a 03 9b 00 32 32  7e
6da8 : 01 9b 00 64 05 00 a0 00  d6
6db0 : 32 32 01 a0 00 32 05 00  f5
6db8 : a5 00 32 32 01 a5 00 64  36
6dc0 : 46 00 eb 00 62 03 01 aa  99
6dc8 : 00 62 41 00 aa 00 60 03  7c
6dd0 : 01 aa 00 60 41 00 eb 00  f6
6dd8 : 5e 03 01 aa 00 5e 41 00  45
6de0 : aa 00 4e 10 01 aa 00 4e  22
6de8 : 28 00 d2 00 4c 03 01 aa  fb
6df0 : 00 4c 28 00 aa 00 4a 03  fa
6df8 : 01 aa 00 4a 28 00 d2 00  66
6e00 : 48 03 01 aa 00 48 28 00  42
6e08 : aa 00 38 10 01 aa 00 38  98
6e10 : 41 00 eb 00 36 03 01 aa  21
6e18 : 00 36 41 00 aa 00 34 03  05
6e20 : 01 aa 00 34 41 00 eb 00  c1
6e28 : 32 03 01 aa 00 32 41 00  08
6e30 : f0 00 32 46 00 f0 00 32  62
6e38 : 32 01 f0 00 64 46 00 36  0c
6e40 : 01 62 03 01 f5 00 62 41  bf
6e48 : 00 f5 00 34 2e 01 f5 00  8c
6e50 : 34 41 00 36 01 34 03 01  ab
6e58 : fa 00 36 3c 00 fa 00 36  ac
6e60 : 2a 01 fa 00 60 3c 00 3b  28
6e68 : 01 32 46 00 81 01 32 03  03
6e70 : 01 3b 01 34 46 00 3b 01  29
6e78 : 34 03 01 3b 01 36 46 00  b0
6e80 : 81 01 36 03 01 65 01 38  20
6e88 : 1b 00 65 01 38 2c 01 60  c7
6e90 : 01 64 05 00 60 01 38 2c  4c
6e98 : 01 5b 01 38 05 00 5b 01  4e
6ea0 : 38 2c 01 56 01 64 05 00  41
6ea8 : 56 01 38 2c 01 86 01 32  bf
6eb0 : 46 00 cc 01 32 32 01 86  0f
6eb8 : 01 64 46 00 86 01 34 30  1f
6ec0 : 01 86 01 34 41 00 c7 01  00
6ec8 : 34 2e 01 8b 01 62 3c 00  d9
6ed0 : 8b 01 36 2c 01 8b 01 36  cc
6ed8 : 37 00 c2 01 36 2a 01 90  ba
6ee0 : 01 60 32 00 90 01 38 28  e0
6ee8 : 01 90 01 38 2d 00 d6 01  a9
6ef0 : 32 32 01 d6 01 64 05 00  9e
6ef8 : db 01 32 32 01 db 01 32  7e
6f00 : 05 00 e0 01 32 32 01 e0  d8
6f08 : 01 64 05 00 e5 01 36 2e  18
6f10 : 01 e5 01 36 23 00 08 02  61
6f18 : 36 0f 01 ea 01 45 1e 00  26
6f20 : ea 01 45 03 01 ea 01 47  37
6f28 : 23 00 0d 02 34 14 01 e5  83
6f30 : 01 34 28 00 e5 01 32 03  8b
6f38 : 01 e5 01 32 2d 00 12 02  d2
6f40 : 32 18 01 ea 01 49 28 00  17
6f48 : ea 01 49 03 01 ea 01 4b  68
6f50 : 0f 00 f9 01 4b 19 01 f9  73
6f58 : 01 64 05 00 fe 01 4b 05  fc
6f60 : 00 fe 01 4b 19 01 03 02  33
6f68 : 4b 1a 01 1c 02 32 46 00  4f
6f70 : 1c 02 32 03 01 1c 02 34  dc
6f78 : 46 00 62 02 34 03 01 1c  2f
6f80 : 02 36 46 00 1c 02 36 17  08
6f88 : 01 1c 02 4d 05 00 21 02  9a
6f90 : 38 15 01 21 02 38 05 00  ad
6f98 : 26 02 38 15 01 26 02 4d  54
6fa0 : 37 00 5d 02 4b 03 01 2b  96
6fa8 : 02 4b 32 00 2b 02 49 03  ca
6fb0 : 01 2b 02 49 37 00 62 02  f2
6fb8 : 49 1b 01 5d 02 64 05 00  d2
6fc0 : 5d 02 4f 15 01 58 02 4f  0e
6fc8 : 05 00 58 02 4f 15 01 1c  fe
6fd0 : 02 64 3c 00 1c 02 62 03  75
6fd8 : 01 1c 02 62 37 00 53 02  79
6fe0 : 60 03 01 1c 02 60 37 00  86
6fe8 : 00 00 69 ff 00 ff 00 69  15
6ff0 : ff 00 fe 01 69 82 00 00  7a
6ff8 : 00 6b ff 00 ff 00 6b ff  5b
7000 : 00 fe 01 6b 82 00 00 00  55
7008 : 6d ff 00 ff 00 6d ff 00  e0
7010 : fe 01 6d 82 00 7f 02 69  11
7018 : 03 01 00 00 6b 03 01 9f  ae
7020 : 01 00 05 00 a1 01 00 03  8b
7028 : 01 a5 01 00 03 01 a6 01  11
7030 : 00 02 03 a8 01 00 02 02  23
7038 : a9 01 00 03 01 ff ff 00  d2
7040 : a0 ff 85 5c 86 5d c8 b1  a7
7048 : 5c 85 6d c8 b1 5c 85 a3  37
7050 : 30 44 c8 b1 5c 85 6e c8  48
7058 : b1 5c 85 6f c8 b1 5c d0  b4
7060 : 1a 20 bf 71 48 a5 6d a6  5d
7068 : a3 a4 6e 20 dd 62 18 69  21
7070 : 01 d0 01 e8 c6 6f d0 f3  4a
7078 : 4c 91 70 c9 01 d0 18 20  19
7080 : bf 71 48 a5 6d a6 a3 a4  a2
7088 : 6e 20 dd 62 c8 c6 6f d0  ec
7090 : f8 68 a8 4c 46 70 60 c9  6d
7098 : 02 d0 1b 20 bf 71 48 a5  c1
70a0 : 6d a6 a3 a4 6e 20 dd 62  02
70a8 : 38 e9 01 b0 01 ca c8 c6  42
70b0 : 6f d0 f2 4c 91 70 20 bf  6a
70b8 : 71 48 a5 6d a6 a3 a4 6e  5b
70c0 : 20 dd 62 18 69 01 90 01  4d
70c8 : e8 c8 c6 6f d0 f2 4c 91  ad
70d0 : 70 85 5c 86 5d a0 00 8c  df
70d8 : 10 5f b1 5c d0 01 60 c8  b8
70e0 : 8c 0e 5f c9 01 d0 0f b1  bb
70e8 : 5c 8d 09 5f c8 b1 5c 8d  e0
70f0 : 0a 5f c8 4c da 70 c9 02  c2
70f8 : d0 0b ad 10 5f 49 ff 8d  17
7100 : 10 5f 4c da 70 c9 c0 90  a8
7108 : 03 38 e9 c0 48 ad 0a 5f  93
7110 : 0a 0a 0a 8d 0d 5f a9 00  c6
7118 : 8d 0c 5f ad 09 5f 0a 2e  49
7120 : 0c 5f 0a 2e 0c 5f 0a 2e  64
7128 : 0c 5f 8d 0b 5f ae 0c 5f  03
7130 : ac 0d 5f 20 da 62 a9 08  b6
7138 : 8d 0f 5f a9 00 85 5b 68  c4
7140 : 0a 26 5b 0a 26 5b 0a 26  27
7148 : 5b 85 5a a5 5b 18 69 d8  7f
7150 : 85 5b a0 00 a2 12 a5 21  3f
7158 : 20 ce 62 e8 a5 20 20 ce  0e
7160 : 62 ad 00 ff aa 09 01 8d  ab
7168 : 00 ff b1 5a 8e 00 ff 2c  61
7170 : 10 5f 10 03 20 ce 71 a2  18
7178 : 1f 20 ce 62 a5 21 a2 12  b9
7180 : 20 ce 62 a5 20 18 69 50  5e
7188 : 85 20 a5 21 69 00 85 21  9a
7190 : c8 c0 08 d0 bf 2c 10 5f  31
7198 : 10 09 ad 09 5f 18 69 03  1c
71a0 : 8d 09 5f ee 09 5f ad 09  bc
71a8 : 5f c9 50 90 08 e9 50 8d  3e
71b0 : 09 5f ee 0a 5f ac 0e 5f  b8
71b8 : 20 bf 71 a8 4c da 70 98  b8
71c0 : 18 65 5c 85 5c a5 5d 69  8e
71c8 : 00 85 5d a9 00 60 8d 11  73
71d0 : 5f a2 08 8c 12 5f 0e 11  8a
71d8 : 5f 08 a0 03 28 08 26 62  e4
71e0 : 26 61 26 60 26 5f 88 10  ec
71e8 : f3 28 ca d0 e9 a0 00 a5  ab
71f0 : 21 a2 12 20 ce 62 a5 20  c2
71f8 : e8 20 ce 62 a2 1f b9 5f  b9
7200 : 00 20 ce 62 a2 12 20 ce  e9
7208 : 62 e6 20 d0 02 e6 21 c8  6d
7210 : c0 03 d0 db a5 21 a2 12  13
7218 : 20 ce 62 a5 20 e8 20 ce  54
7220 : 62 a5 20 38 e9 03 85 20  71
7228 : a5 21 e9 00 85 21 b9 5f  df
7230 : 00 ac 12 5f 60 8e c1 5f  37
7238 : 8c c2 5f a2 01 8e c0 5f  98
7240 : e8 8e c3 5f 48 4a 4a 4a  e1
7248 : 4a f8 aa f0 08 18 a9 00  bf
7250 : 69 16 ca d0 fb 8d 1a 5f  e4
7258 : d8 68 29 0f c9 0a 90 02  c4
7260 : 69 05 f8 6d 1a 5f d8 48  c8
7268 : 4a 4a 4a 4a 09 30 c9 30  4d
7270 : d0 02 a9 20 8d c4 5f 68  fd
7278 : 29 0f 09 30 8d c5 5f a9  49
7280 : 00 8d c6 5f a9 c0 a2 5f  ce
7288 : 4c d1 70 08 8e c1 5f 8c  67
7290 : c2 5f a2 01 8e c0 5f e8  09
7298 : 8e c3 5f 28 f0 04 a0 00  96
72a0 : f0 02 a0 04 a2 00 8e 19  d1
72a8 : 5f 98 aa bd 87 6b ae 19  76
72b0 : 5f 9d c4 5f c8 e8 e0 04  5a
72b8 : d0 ec a9 00 8d c8 5f a9  59
72c0 : c0 a2 5f 4c d1 70 4a 90  1e
72c8 : 04 a2 2e d0 02 a2 00 0a  0c
72d0 : 18 69 10 a8 60 20 d8 72  05
72d8 : a0 80 a2 12 98 20 ce 62  2e
72e0 : a2 1f a9 00 20 ce 62 a2  c3
72e8 : 1e 20 ce 62 88 10 eb a2  14
72f0 : 19 a9 80 0d 30 5f 20 ce  bc
72f8 : 62 60 a9 05 a2 73 20 d1  7f
7300 : 70 20 80 69 60 01 1f 0b  6e
7308 : 20 20 20 20 20 20 20 20  08
7310 : 20 20 20 20 20 20 20 20  10
7318 : 20 20 01 1f 0c 20 20 52  54
7320 : 55 4e 44 45 20 20 42 45  ed
7328 : 45 4e 44 45 54 20 20 01  17
7330 : 1f 0d 20 20 20 20 20 20  a6
7338 : 20 20 20 20 20 20 20 20  38
7340 : 20 20 20 20 00 01 16 0c  f5
7348 : 02 47 41 4d 45 20 4f 56  27
7350 : 45 52 00 ad 3d 5f 18 69  76
7358 : 07 6d 12 d0 29 0f 8d 3d  70
7360 : 5f aa bd 74 73 f0 ec 8a  9a
7368 : 0a a8 b9 85 73 aa b9 84  62
7370 : 73 4c 40 70 01 01 01 01  46
7378 : 01 01 01 01 01 01 00 00  72
7380 : 00 00 00 00 56 69 a4 73  ab
7388 : e5 73 58 74 99 74 b2 74  bc
7390 : fd 74 20 75 75 75 ed 75  24
7398 : 00 00 00 00 00 00 00 00  99
73a0 : 00 00 00 00 b9 00 28 f0  bf
73a8 : 00 b9 00 28 1e 03 a9 01  2d
73b0 : 28 1e 03 d7 00 46 f0 00  99
73b8 : b9 00 28 5a 01 d7 00 46  22
73c0 : 5a 01 c7 01 46 5a 01 b9  5c
73c8 : 00 82 1e 03 d7 00 a0 f0  d3
73d0 : 00 a9 01 28 5a 01 a9 01  41
73d8 : 82 1e 03 b9 00 82 f0 00  39
73e0 : ff ff ff ff ff c8 00 46  b1
73e8 : 32 00 c8 00 46 14 02 b4  c3
73f0 : 00 5a 19 01 b4 00 73 14  c5
73f8 : 03 c8 00 87 32 00 0e 01  ae
7400 : 73 14 02 fa 00 5f 14 03  af
7408 : c8 00 5f 32 00 c8 00 5f  f3
7410 : 14 02 2c 01 46 2d 01 2c  7b
7418 : 01 73 4b 00 5e 01 5a 2d  57
7420 : 01 90 01 46 14 02 a4 01  58
7428 : 78 28 00 c2 01 6e 0a 03  be
7430 : ae 01 6e 0a 02 ae 01 6e  b2
7438 : 14 00 ae 01 6e 14 00 a4  e9
7440 : 01 78 0f 03 b3 01 87 14  2b
7448 : 00 e0 01 6e 19 01 ea 01  0e
7450 : 6e 0a 02 ff ff ff ff ff  44
7458 : 40 01 64 3c 03 40 01 64  b8
7460 : 3c 02 04 01 a0 78 00 46  19
7468 : 00 14 78 00 46 00 14 3c  be
7470 : 01 50 00 5a 64 00 be 00  26
7478 : 14 3c 01 c2 01 14 78 00  d6
7480 : c2 01 14 3c 01 3a 02 14  62
7488 : 3c 01 3a 02 14 3c 01 cc  d4
7490 : 01 5a 64 00 ff ff ff ff  d7
7498 : ff 25 01 14 a0 02 5a 01  72
74a0 : 14 a0 03 25 01 64 50 02  e2
74a8 : 5a 01 64 50 03 ff ff ff  d6
74b0 : ff ff 00 00 62 ff 00 ff  d5
74b8 : 00 62 37 00 00 00 66 ff  51
74c0 : 00 ff 00 66 37 00 4a 01  2b
74c8 : 62 ff 00 49 02 62 37 00  63
74d0 : 4a 01 66 ff 00 49 02 66  53
74d8 : 37 00 18 01 5a 50 00 18  8e
74e0 : 01 6e 50 00 50 00 0a 3c  d2
74e8 : 02 30 02 0a 3c 03 14 00  f0
74f0 : 82 3c 03 6c 02 82 3c 02  08
74f8 : ff ff ff ff ff 02 00 32  6b
7500 : c6 00 00 00 96 c8 00 c8  08
7508 : 00 32 64 01 b7 01 32 c6  34
7510 : 00 b7 01 96 c8 00 b7 01  6c
7518 : 32 64 01 ff ff ff ff ff  bd
7520 : 36 01 14 a0 01 4a 01 14  7f
7528 : a0 01 36 01 14 14 00 36  45
7530 : 01 b4 14 00 18 01 14 14  93
7538 : 00 18 01 14 50 01 54 01  68
7540 : 14 14 00 54 01 14 50 01  dd
7548 : 2c 01 14 50 01 68 01 14  83
7550 : 50 01 68 01 64 46 03 54  88
7558 : 01 64 50 03 18 01 64 46  a8
7560 : 02 2c 01 64 50 02 d2 00  a6
7568 : aa 0a 03 ae 01 aa 0a 02  40
7570 : ff ff ff ff ff c8 00 28  05
7578 : 73 00 45 01 28 73 00 c8  0d
7580 : 00 a0 73 00 45 01 a0 73  73
7588 : 00 c8 00 00 a0 01 b8 01  e3
7590 : 28 a0 01 00 00 6e a0 00  3f
7598 : e0 01 5a a0 00 a0 00 14  d1
75a0 : 5a 01 e0 01 5a 5a 01 14  78
75a8 : 00 14 8c 00 e0 01 b4 8c  d7
75b0 : 00 00 00 28 8c 00 14 00  cf
75b8 : 3c 8c 00 00 00 50 8c 00  ef
75c0 : 00 00 50 8c 00 f4 01 a0  53
75c8 : 8c 00 e0 01 8c 8c 00 f4  c4
75d0 : 01 78 8c 00 a0 00 6e 3c  6d
75d8 : 02 c8 00 64 50 02 08 02  04
75e0 : 14 50 02 1c 02 1e 3c 02  27
75e8 : ff ff ff ff ff 28 00 5a  dd
75f0 : ff 00 27 01 5a ff 00 26  cb
75f8 : 02 5a 33 00 28 00 6e ff  30
7600 : 00 27 01 6e ff 00 26 02  3f
7608 : 6e 33 00 28 00 14 ff 00  b6
7610 : 27 01 14 ff 00 26 02 14  1e
7618 : 32 00 28 00 b4 ff 00 27  ee
7620 : 01 b4 ff 00 26 02 b4 33  27
7628 : 00 28 00 6e 46 01 28 00  17
7630 : 14 46 01 58 02 14 46 01  8f
7638 : 58 02 6e 46 01 ff ff ff  06
7640 : ff ff 00 00 00 00 00 00  3f
```

Listing 1. »Vectors« (Schluß)

C128 - Programmieren in Maschinensprache

Dieses Werk von Gerd Möllmann, seines Zeichens »Vater« von Hypra-Ass und Top-Ass, ist kein Lehrbuch zur 6502/6510-Assembler-Programmierung, sondern vielmehr weiterführende und vertiefende Literatur, die dem 6502-Vertrauten den Weg zur effektiven Assembler-Programmierung mit dem C128 ebnen soll.

Das Buch ist in sieben Kapitel gegliedert, die jedoch keinen Lehrbuch-Charakter aufkommen lassen. Diese Struktur macht das Buch zu einem guten Nachschlagewerk, das als kompetenter Ratgeber zu allen Bereichen der Programmierung des C128 in jeder Bibliothek eines C128-Besitzers stehen sollte.

Im ersten Kapitel werden die Bausteine des C128 (MMU, VDC, VIC, SID und CIAs) behandelt. Das zweite Kapitel beschreibt die Routinen der »Common Area«, also diejenigen Kernel- und Interpreter-Unterprogramme, die im RAM-Bereich von $0000 bis $03FF liegen. Im dritten Kapitel wird auf 50 Seiten das Betriebssystem analysiert, daran angehängt die Kernel-Vektoren.

Im fünften und längsten Kapitel wird der Basic-Interpreter sorgfältig zerlegt. Das sechste Kapitel beschäftigt sich mit den dazugehörigen Basic-Vektoren in der erweiterten Zeropage und im letzten Kapitel werden schließlich deren Einsatzmöglichkeiten beschrieben.



Ein ausführliches Nachschlagewerk für die Assembler-Programmierung des C128, mit guten Beispielen und übersichtlich gestaltet.

(Jörg Sahlmann/sk)

Gerd Möllmann: C128-Programmieren in Maschinensprache, Markt & Technik Verlag, 270 Seiten, ISBN 3-89090-213-8, 52 Mark



Vom C64 zum C128 - Tips & Tricks

Dieses Buch ist einerseits ein fantastisches Rüstzeug für die vielen C64-Umsteiger, kann aber gleichermaßen allen anderen C128-Besitzern empfohlen werden.

Das Kapitel »Tastatur und Editor« eignet sich vor allem für Einsteiger.

Ein weiteres, sehr umfangreiches Kapitel, befaßt sich mit Basic 7.0 - vor allem für Fortgeschrittene. Es enthält viele Utilities und informative wie verständliche Programmierkurse für Windows, Menüs und Ein-/Ausgabe-Masken. Außerdem wird das Umsetzen von C64-Programmen auf den C128 ausführlich beschrieben. Ganz besonders überzeugen interessante Grafik-Beispielprogramme, wie »Apfelberge« und »Roulette«.

Das Kapitel »Maschinensprache« beschreibt die wichtigsten Arbeitsmittel und das C128-Betriebssystem sowie die Adaption von C64-Programmen. Im fünften Kapitel werden alle Unterschiede des C64-Modus zum »normalen« C64 aufgezeigt.

»Einstieg in CP/M« ermöglicht einen leichten Einstieg in diesen Bereich und enthält viele Tips und Tricks. Das letzte Kapitel - »Diskettenlaufwerke« - erklärt in vergleichender Form die Programmierung und Anwendung der Floppies 1541 und 1570/1571 und enthält viele Praxistips.

Alle abgedruckten Programme sind auch auf der dem Buch beiliegenden Diskette.

Fazit: Ein »Muß« für alle C128-Anwender, die mit dem C128 bequemer arbeiten und sich gleichzeitig weiterbilden wollen.

(K. Ruprecht/M. Jobst)

Florian Müller: Vom C64 zum C128 - Tips & Tricks, Markt & Technik, 2. Auflage 1987, 290 Seiten inklusive Beispiel- und Utility-Diskette, ISBN 3-89090-402-5, 49 Mark

C128 ROM-Listing

Ein unentbehrliches Hilfsmittel für Maschinensprache-Programmierer ist das ROM-Listing. Dieses Buch aus der Commodore-Sachbuchreihe ist das erste vollständige ROM-Listing zum C128, das die gesamten 44-KByte-ROM, also auch den 28 KByte langen Basic-7.0-Interpreter, umfaßt.

Es erklärt zunächst die komplizierte Speicherverwaltung des C128, also auch die Technik des »Bankswitching«.

Des weiteren werden die Videocontroller VIC und VDC, der Soundchip SID, sowie die CIAs in Kurzform behandelt und alle vom System beeinflußten Adressen aufgeführt.

Den Hauptteil macht mit 375 Seiten das ROM-Listing selbst aus. Die äußere Struktur des ROM-Listings besticht durch Klarheit und Übersichtlichkeit, da es sich um einen Quelltext-Ausdruck handelt, und die eingesetzten Label-Adreßangaben eine schnelle Suche von



Sprungzielen ermöglichen. Auch bei den Kommentaren ist das Konzept gut geglückt: Trotz ausführlicher Erläuterungen wird das ROM-Listing keinesfalls unübersichtlich.

Fazit: Das vorliegende Buch ist für C128-Maschinenprogrammierer sehr empfehlenswert. Aufgrund des enormen Umfangs (456 Seiten), des großen Nutzens und eines guten Schlagwortregisters ist der Preis von 58 Mark angemessen.

Was vielleicht noch wünschenswert gewesen wäre, ist eine prinzipielle Einführung in die Arbeit mit einem ROM-Listing und die praktische Anwendung eines solchen.

(Florian Müller/M. Jobst)

Commodore-Sachbuchreihe, Dr. Ruprecht: C128 ROM-Listing, Markt & Technik Verlag, 456 Seiten, ISBN 3-89090-212-X, 58 Mark



Programmieren in Assembler mit Top-Ass-Plus für den C128

Maschinensprache zu erlernen, ist der Traum jedes Programmierers; ein guter Assembler ist der Wunsch aller, die bereits Maschinensprache beherrschen. Dieses umfassende Werk wird beiden Ansprüchen gerecht.

Ein über 100 Seiten langes Kapitel zur Erlernung der 65xx-Maschinensprache im C128-Modus vermittelt dem Leser das Hexadezimal- und Binärsystem, alle Adressierungsarten und Befehle, sowie Kenntnisse über allgemeine Programmiertechniken und TEDMON, den in der Firmware eingebauten Maschinensprache-Monitor des C128.

Danach hat der Leser das nötige Wissen, um selbst kleine Maschinenprogramme zu schreiben. Er verfügt obendrein über ein hervorragendes Nachschlagewerk mit zahlreichen Tabellen, die eine große Hilfe beim Programmieren darstellen.

Die Anwendung des Gelernten ist Gegenstand des zweiten Kapitels, welches die Speicherverwaltung, die Betriebssystemroutinen sowie die Ein-/Ausgabe-Steuerung des C128 beschreibt.

Der restliche Teil des Werkes erläutert sehr ausführlich und praxisnah den Umgang mit der auf Diskette beiliegenden Top-Ass-Software. Auf diese wird auch noch an anderer Stelle in diesem Sonderheft eingegangen (»Software im Überblick«, Seite 155).

Im Anhang findet man noch Tabellen, die sich auf das Top-Ass-Software-Paket und den Befehlssatz des Prozessors beziehen.

(Florian Müller/M. Jobst)

Gerd Möllmann, Michael Bauer: Programmieren in Assembler mit Top-Ass Plus/C128, Markt & Technik, 367 Seiten inkl. Diskette, ISBN 3-89090-416-5, 59 Mark

Das Dritte im Bunde

Nun halten Sie bereits das dritte Sonderheft in Ihren Händen, das sich ausschließlich mit dem C128 und seinen Anwendungen beschäfigt. Doch die vorangegangenen Ausgaben zum Thema C128 besitzen auch weiterhin Aktualität, wie Sie in folgendem Artikel lesen können.

Wer vor etwa zwei Jahren den sensationellen neuen C128 erwarb, wurde nach der ersten Euphorie über diesen neuartigen Computer von einer bedauernswerten Tatsache überrascht. Für den C128-Modus, der eigentlich leistungsfähigen Betriebsart dieses Computers, gab es anfänglich kaum Software. Bisher wagte sich kein Programmierer an den noch unbekannten C128 heran.

Fehlten noch professionelle Anwenderprogramme wie Textverarbeitung und Programmiersprachen sowie die für Entspannung sorgenden Spiele, mußte der betrübte C128-Anwender seinen Computer zunächst vornehmlich im C64-Modus betreiben, wo ihm eine umfangreiche Palette an Software zur Verfügung stand. Mutige Besitzer erprobten auch den CP/M-Modus ihres Computers, der die Welt der professionellen CP/M-Software erschloß.

Das Ende der »Software-Krise«

Der C128-Modus mit seinem leistungsfähigem Basic mußte weiterhin auf seinen Einsatz warten. Hier war der Hobbyprogrammierer auf seine eigenen Fähigkeiten angewiesen. Doch gerade dieser akute Mangel an Software beflügelte manchen Anwender, so daß das 64'er-Magazin bald eine eigene 128'er-Rubrik einrichten konnte, in denen C128-Besitzer ihre Erfahrungen mit dem Computer weitergaben. Zu Beginn des Jahres 1986 wurde sodann das erste C-128-Sonderheft zusammengestellt. Der Anfang einer Serie, von der Sie heute den dritten Teil in Form eines weiteren 128er-Sonderheftes in den Händen halten. Es bietet im Schwerpunkt viele Listings, die aus der Feder von C128-Hobbyprogrammierern stammen, und den C128-Modus perfekt nutzen.

Doch auch die beiden Vorgänger dieses Sonderheftes, Sonderheft 1/86 und 10/86 (Bild 1), können sowohl für Einsteiger als auch Fortgeschrittene Programmierer aktuell sein. Ein kleiner Rückblick, den wir nun starten wollen, soll dies beweisen.

Recht früh, gleich zu Beginn der C128-Ära im Januar 1986 konnte die 64'er-Radaktion bereits das erste Sonderheft herausgeben, das sich nur mit dem C128 beschäftigte (Bild 1, links). Es war der ideale Ratgeber für die damals meist ratlosen Einsteiger, die sich mit dem umfangreichen, aber nicht optimalen Handbuch des C128 abmühen mußten. Mit Gewicht auf die für das Programmieren notwendigen Grundlagen bot es dem Neuling einen tiefen Einblick in das von ihm erworbene Computersystem, der auch dem heutigen Einsteiger von Nutzen ist.

Ein Rundgang durch die Hardware des C128 führt Sie in die elementare Welt des Computers ein. Ausgehend vom User-Port des C128 begeben Sie sich auf eine Wanderung quer über die große Computer-Platine, die ja eigentlich drei verschiedene Computer enthält. Bald zeigt sich das zunächst verwirrende Innenleben auf der Platine als logisches Zusammenspiel von komplexen Bausteinen und Gruppen, die den C128 zu dem machen, was er ist.

Schaltet man nun das nächste Mal seinen Computer ein, wird das Gerät, mit dem man programmieren kann, keine »Black Box« mehr sein, in der unbekannte Vorgänge ablaufen. Der C128 wird »durchsichtiger« und überschaubarer mit jeder Minute, die Sie am Monitor arbeiten.

Monitore sind auch ein weiteres Thema des Sonderheftes 1/86, denn für die ordnungsgemäße Arbeit mit dem C128 ist ein scharfes Bild sehr wichtig. Obgleich man einen handelsüblichen Fernseher an seinen Computer anschließen kann, so bietet er doch nicht das angenehme Bild eines guten Monitors. Zudem ist eine 80-Zeichen-Darstellung



Bild 1. Die beiden schon für den C128 erschienenen Sonderhefte 1/86 (links) und 10/86 (rechts)

hier kaum denkbar. Aus diesem Grund gibt ein ausführlicher Test Auskunft über die Vor- und Nachteile der besten Monitore für den C128. Eine Vorstellung aller gängigen Kabelverbindungen und Buchsen bringt zusätzlich Licht in die bei Monitorherstellern üblichen Stecker-Normen, so daß man sich stets einer richtigen Verbindung von Computer und Bildschirm gewiß sein kann. Eine kleine Bauanleitung, die das Umschalten zwischen 40- und 80-Zeichen-Darstellung vornimmt, rundet das Thema Monitor ab.

Basic 7.0: Der professionelle Basic-Interpreter

Wer nach dem Kauf eines C128 das etwa 300 Seiten starke Bedienungshandbuch durchblättert, wird sich bald der Leistungsfähigkeit des Basic 7.0 bewußt, das sich im C128-Modus präsentiert. Doch mit dem Erlernen der vielen Befehle, wie es das Handbuch vermittelt, ist es noch nicht getan. Noch fehlt das Wissen darum, wie man die Kommandos in Verbindung mit anderen Befehlen wirken läßt. Es mangelt an Programmiererfahrung, die für das Erzielen von befriedigenden Ergebnissen auf dem Computer

äußerst wichtig ist. Hier greift eine Reihe von Berichten des Sonderheftes 1/86 ein.

Wird man zunächst über grundlegende Eigenschaften des C128 informiert, befaßt sich ein weiterer Artikel mit der genauen Analyse des hervorragenden Basic. Übersichtlich und lückenlos werden alle Befehle des Basic 7.0 erläutert. Ergeben sich dabei Besonderheiten, so werden diese dem Leser natürlich auch mitgeteilt.

Nachdem man sich mit den Befehlen des C128 vertraut gemacht hat, kann man sogleich Programmiererfahrung sammeln, wenn man einen weiteren Bericht studiert. Hier trifft man auf den Begriff »strukturiertes Programmieren«, der unlängst nur höheren Sprachen wie Pascal oder C vorbehalten war.

Spezielle Befehle und Erweiterungen erlauben es dem C128-Programmierer, neue und übersichtliche Programmstrukturen aufzubauen, wie man sie von Pascal her kennt. Mit ihrer Hilfe können verwirrende Programmsprünge in Zeilennummern und viele andere zum gefürchteten »Spaghetticode« führende Befehls-Konstruktionen vermieden werden. So gestattet die Erweiterung des IF-THEN-Statements durch den kleinen Begriff ELSE eine wesentlich flexiblere Handhabung der bedingten Verzweigung. Befehle wie BEGIN und BEND fügen zusammengehörige Anweisungen zu einem übersichtlichen Block und die neuartigen Schleifen-Kommandos DO-LOOP, WHILE und UNTIL ermöglichen einen ähnlich übersichtlichen Aufbau wie in Pascal.

Sprites und Shapes im Griff

Das Geheimnis vieler Spiele auf dem C64 ist bei den kleinen beweglichen Grafiken namens »Sprites« zu suchen, die sich stets in neuen Gestalten auf dem Bildschirm tummeln. Während das Programmieren und Steuern der Sprites mit dem C64 in Basic sehr umständlich ist, und wohl den Assembler-Profis vorbehalten scheint, ist der C128 mit vielen Befehlen und sogar einem integrierten Spriteeditor ausgerüstet, was das Erstellen und Lenken von Sprites vereinfacht. Ein Kurs in Sonderheft 1/86 befaßt sich deshalb ausführlich mit Sprites und deren Eigenschaften. Doch geht die Themenstellung weit über die Erläuterung des Spriteeditors und die zur Verfügung stehenden Steuerbefehle hinaus. Anhand von praktischen Beispielen wird man in die professionelle Spriteprogrammierung eingeführt. Interessante Begriffe sind hier die Animation von Sprites oder die Erzeugung und Koordination von »großen« Sprites durch Zusammensetzen vieler kleiner Sprites. Besondere Aufmerksamkeit wird dabei den komfortablen Kollisions-Befehlen, wie zum Beispiel COLLISION und BUMP, geschenkt, die den Zusammenstoß von Sprites mit Hintergrundzeichen oder anderen Sprites ermitteln.

Daneben beschäftigt sich der Kurs mit einer weiteren Grafikeigenschaft des C128, die man beim C64 vergeblich sucht: den Shapes.

Shapes sind wie Sprites Grafikausschnitte. Sie sind jedoch nicht ganz so agil. Mit ihrer Hilfe lassen sich Bereiche des hochauflösenden Grafikbildschirms in Stringvariablen ablegen, um sie danach wieder an anderer Stelle in das Grafikbild zurückzukopieren. Dabei kann man zwischen verschiedenen Abbildungsmodi wählen. Neben dem direkten Kopieren lassen sich Shapes additiv oder subtraktiv mit den Grafikpunkten des betreffenden Bereiches überlagern. Sie können mit einigen Tricks auch auf Diskette oder Kassette gespeichert werden. Selbst das Umwandeln von Shapes in Sprites ist durchaus möglich. Welche Grafikvielfalt sich dadurch ergibt, wird im Spritekurs ausführlich beschrieben.

Neben all diesen Grundlagen findet man selbstverständlich auch viele Listings zum Abtippen, die in der Software-Sammlung eines C128-Besitzers nicht fehlen sollten. Besonders interessant ist ein Grafikprogramm, das die Grafikbefehle des C128 um eine wunderbare Eigenschaft erweitert.

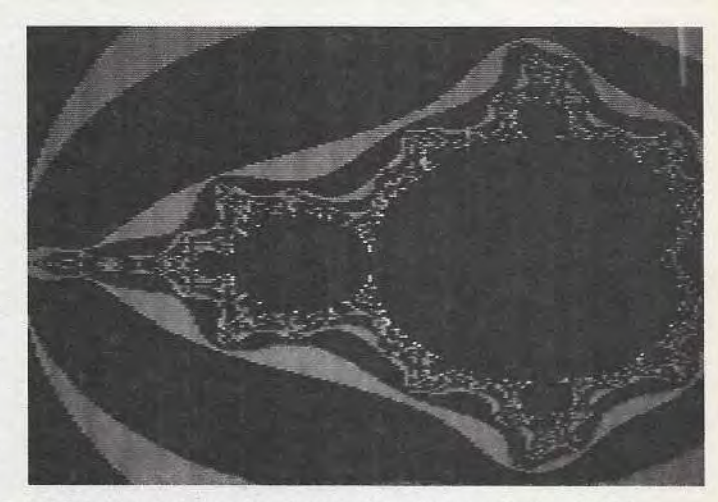


Bild 2. Faszinierende »Apfelmännchen« mit dem C128. Ein Programm in Sonderheft 1/86 macht's möglich.

Wie Sie vielleicht wissen, ist der VDC-Chip, der für den Aufbau des 80-Zeichen-Bildschirms zuständig ist, fähig, auch Einzelpunkt-Grafiken auf dem 80-Zeichen-Monitor zu verwalten. Leider unterstützen die umfangreichen Grafikbefehle nur den 40-Zeichen-Modus bei einer Auflösung von 320 mal 200 Punkten. Im 80-Zeichen-Grafikmodus, der die doppelte Punktauflösung von 640 mal 200 Punkten bietet, sind sie nicht funktionsfähig.

Anders jedoch, wenn Sie jenes Grafikprogramm mit dem Namen »Grafik-80« besitzen. Nachdem es im Speicher des C128 installiert wurde, stehen von nun an alle bekannten Grafik-Anweisungen wie DRAW, CIRCLE, BOX oder PAINT auch auf dem 80-Zeichen-Bildschirm zur Verfügung. Mit diesem Programm ist man sogar befähigt, simultan mit der 40- und 80-Zeichen-Grafik zu arbeiten, da der erweiterte Interpreter vor jedem Befehl prüft, welcher Bildschirm gerade aktiv ist. Die Befehle werden dann stets im aktuellen Modus ausgeführt.

Fraktales und wie macht man dem C64-Modus Beine?

Nicht minder interessant ist auch ein kleines Basic-Programm, das fantastische Fraktalgrafiken erzeugt. Insbesondere die Mathematiker unter den Grafik-Ästheten wissen, daß solche Grafiken auch »Apfelmännchen« genannt werden, und einer mathematischen Theorie entspringen. Grafiken, die die hauchdünne Grenze zwischen Ordnung und Chaos auf faszinierende Weise sichtbar machen (Bild 2). Ein ausführlicher Bericht in Sonderheft 1/86 zeigt, was sich hinter diesen Formen gebrochener, eben fraktaler Dimensionen verbirgt (lateinisch »frangere« bedeutet »brechen«).

Eine Fortsetzung von Wissenswertem über den C128 kann man in Sonderheft 10/86 (Bild 1, rechts) finden, das den Leser wiederum mit interessanten Informationen und Listings fesselt.

Was man beim C64 oftmals sehnlichst vermißte, ist für den C128-Anwender eine Selbstverständlichkeit. TEDMON, der integrierte Maschinensprache-Monitor, ist sofort

nach dem Einschalten des C 128-Modus aktiv und kann mit einem eigenen Befehl (MONITOR) aufgerufen werden. Damit stehen dem Programmierer wertvolle Hilfen zur Verfügung, um etwa einen bestimmten Bereich des Speichers aufzulisten, nach besonderen Byte-Kombinationen zu suchen oder gar beliebige Bereiche an eine andere Stelle des Arbeitsspeichers zu verschieben. Zusätzlich können Teile des Speichers mit anderen verglichen werden oder Maschinenprogramme geladen oder gespeichert werden.

TEDMON – ein Monitor für alle Fälle

TEDMON besitzt sogar einen kleinen Assembler, der die Eingabe von Maschinenbefehlen erlaubt und diese automatisch in die für den Computer verständlichen »Op-Codes« umwandelt. Ein entsprechender Disassembler wandelt, als Pendant dazu, jene Codes wieder in Assembler-Befehle um, so daß man die Inhalte eines jeden Speicherbereiches als Maschinenprogramm betrachten kann. Die Maschinensprache-Kenner unter den C 128-Besitzern werden TEDMON bereits zu schätzen wissen. Wenige wissen jedoch, daß der Monitor auch für den Basic-Programmierer interessant sein kann. Aus diesem Grund befaßt sich ein spezieller Kurs ausführlich mit den Eigenschaften

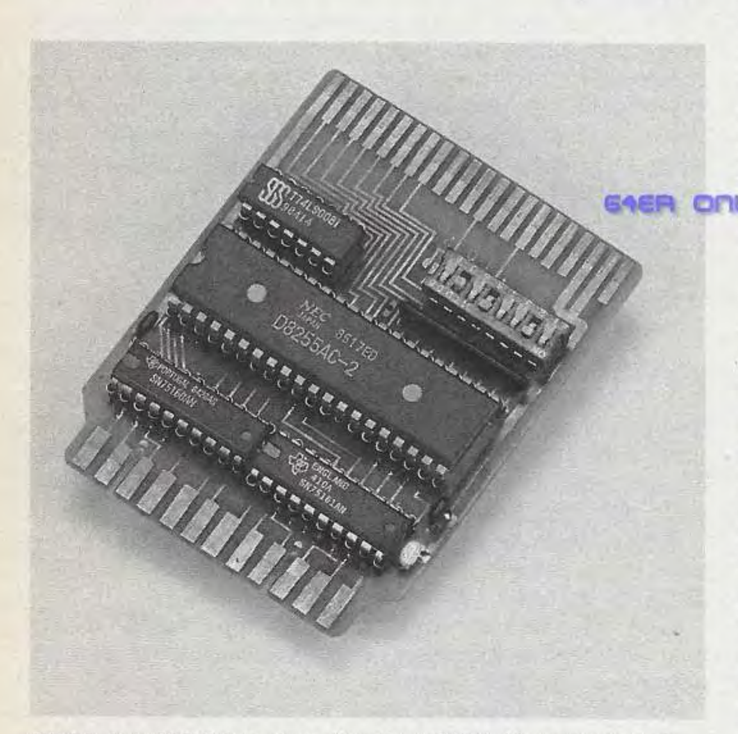


Bild 3. Das IEEE-Interface für den C 128 mit enormen Leistungen. Die Bauanleitung finden Sie in Sonderheft 10/86.

von TEDMON und gibt Tips für seine effektive Nutzung bei der Basic-Programmierung.

Wird man zunächst Schritt für Schritt mit den Kommandos des Monitors vertraut gemacht, begibt man sich sodann in die Welt der professionellen Programmierung der Maschinensprache. Keine Angst, hier handelt es sich nicht um eine Assembler-Kurs. Es werden vielmehr Wege aufgezeigt, wie man mit TEDMON Maschinenprogramme bearbeiten kann. Dazu gehört etwa das Suchen nach bestimmten Befehlskombinationen oder das Vergleichen von identisch scheinenden Programmteilen.

Hat man all die grundlegenden Eigenschaften des TEDMON erfahren, wendet sich der Kurs seinem zweiten Hauptthema zu, die Manipulation von Basic-Programmen. Denn hier eröffnen sich dem Anwender Möglichkeiten, die er bisher für undenkbar gehalten hat.

Doch zuvor erfährt er alles über den eigentlichen Aufbau eines Basic-Programmes im Speicher des C 128. Man stellt fest, daß die Befehle nicht Zeichen für Zeichen gespeichert, sondern in Form von besonderen Codes, den »Tokens«, abgelegt werden. Eine Tabelle mit allen Tokens des C 128 hilft dabei, Basic-Programme bald auch über den Monitor lesen zu können. Daneben werden der genaue Aufbau einer Basic-Zeile, die Verkettung mit den anderen Zeilen und die Bedeutung der Zeilennummern erläutert. Mit diesem Wissen lassen sich nun interessante Effekte erzielen.

Was im Normalfall eigentlich nicht möglich ist, wird mit den neuen Manipulationsfähigkeiten Wirklichkeit. Geben Sie doch mehreren Basic-Zeilen die gleiche Zeilennummer, oder verwenden Sie beispielsweise inverse oder blinkende Kommentare in REM-Statements. TEDMON setzt hier keine Grenzen.

Ebenso grenzenlos scheint auch die Grafik des C 128, die mit einer Vielzahl von Befehlen wie DRAW, CIRCLE oder PAINT schnell und problemlos angesteuert werden kann. Ein weiterer Kurs in Sonderheft 10/86 untersucht diese Befehle sehr genau. Mit vielen Skizzen wird die Bedeutung eines jeden Parameters vedeutlicht, so daß man sich sogleich der ungeheueren Flexibilität der Grafik-Kommandos des C 128 bewußt wird. Besitzt man zusätzlich das Programm »Grafik-80« aus Sonderheft 1/86, das übrigens nochmals in Sonderheft 10/86 in modifizierter und verbesserter Version abgedruckt wurde, kann man seine neu gewonnenen Kenntnisse sowohl mit der Grafik des 40-Zeichen-Modus (320 mal 200 Punkte) als auch mit der doppelten Grafikauflösung des 80-Zeichen-Bildschirms (640 mal 200 ansprechbare Punkte) erproben.

Interessantes für Hardware-Spezialisten

Eine größere Flexibiltät des C 128 erreicht auch eine kleine Hardware-Bastelei. Mit wenigen Handgriffen und etwas Geschick mit dem Lötkolben läßt sich das IEEE-Interface aus Sonderheft 10/86 anfertigen, wie Sie es in Bild 3 sehen können. Eingesteckt am Expansion-Port des Computers ist es dem Anwender nun möglich, wesentlich leistungsfähigere Peripheriegeräte, wie etwa Drucker oder Diskettenlaufwerke an den C 128 anzuschließen. Beispiele sind hier die Floppystationen SFD 1001 und CBM 8050, die mit einer Speicherkapazität von jeweils 1 MByte und zweimal 1 MByte die Floppy 1571 bei weitem übertreffen. Zudem läßt der IEEE-Bus eine sehr viel schnellere Datenübertragung zu, und verleiht dem C 128 noch mehr Professionalität.

Besonders interessant für Hardware-Bastler ist schließlich der komplette Schaltplan des C 128, dem zusätzlich einige Tips beigefügt sind. Hier findet man zum Beispiel kleine Schaltungen, um bis zu sieben Betriebssysteme in nur einem EPROM unterzubringen, die absturzfrei und einfach umgeschaltet werden können.

Darüber hinaus bietet das Sonderheft 10/86 wie sein Vorgänger eine Vielzahl an Programmen, Tips und Tricks, die das Programmiererleben erleichtern. Eines davon ist ein wahrer Diener, denn es erweitert das Betriebssystem des C 128 um interessante und teilweise unglaubliche Funktionen, die so manchen Basic-Kenner, aber auch Maschinensprache-Fans in Erstaunen versetzen. Sein Name ist »BUTLER«.

Neu definierte Escape-Sequenzen erlauben das komfortable Editieren eines Basic-Programmes. Man kann dabei beispielsweise zeilen- oder seitenweise vor- und rückwärts

blättern, sowie Teile des Programmes vorübergehend an einen anderen Ort des Speichers übertragen, um sie zu einem späteren Zeitpunkt wieder dem Gesamtprogramm hinzuzufügen.

Der ungewöhnlichste Befehl ist jedoch Z80CALL. So unglaublich es klingen mag, so gestattet er doch, im Basic des C128 Z80-Maschinenprogramme aufzurufen, die prompt ausgeführt werden und nach Beendigung wieder in den C128-Modus zurückkehren. Ein Leckerbissen für Z80-Programmierer.

Die eben erwähnten Funktionen sind nur ein kleine Teil der in BUTLER integrierten Dienstleistungen.

Wesentlich hektischer geht es bei einem Kopierprogramm für die Floppy 1570/71 zu. Denn »Tornado-Copy« hat seinen Namen zurecht verdient. Unter Zuhilfenahme des großen C128-Speichers kopiert dieses Programm eine Diskettenseite in nur 75 Sekunden, wobei nurmehr zwei Diskettenwechsel notwendig sind. Ist die Diskette nicht vollständig beschrieben, genügt bereits ein Wechsel von Original- und Ziel-Diskette.

Neben diesen beiden Beispielen warten noch eine Vielzahl weiterer Hilfsprogramme im Sonderheft 10/86 auf den Leser, wie zum Beispiel ein Centronics-Druckertreiber, der es gestattet, nahezu jeden Drucker mit Centronics-Schnittstelle problemlos anzusteuern.

Hinzu kommt das bereits erwähnte Grafikprogramm »Grafik-80« in einer überarbeiteten Version, der gleich zwei leistungstarke Erweiterungen angefügt wurden. War mit Grafik-80 bisher keine Textdarstellung in der Einzelpunktgrafik möglich, erlaubt nun ein neuer CHAR-Befehl die Beschriftung von Grafiken im VDC-Modus (80-Zeichen-Bildschirm). Anders als mit dem bereits bekannten CHAR-Kommando lassen sich nun die Zeichen punktweise plazieren, wobei sogar das beliebige Versetzen um eine Halbzeile nach oben oder unten möglich ist. Indizes und Potenzen etwa bei der Dokumentation eines mathematischen Graphen sind damit kein Problem mehr.

Die größte Leistung der CHAR-Anweisung liegt jedoch in der Fähigkeit, die auszugebende Zeichenkette in jedem Winkel bezüglich des Anfangspunktes zu schreiben.

So schön eine Grafik auf dem 80-Zeichen-Bildschirm sein kann, so betrüblich war es bisher, daß man das Kunstwerk nicht auf einem Drucker festhalten konnte. Aus diesem Grund ersannen die Autoren von Grafik-80 eine der wohl erfreulichsten Erweiterungen ihres Programms, die Routine »Hardcopy-80«. Der Ausdruck darf dabei in einfacher und doppelter so wie inverser Darstellung vorgenommen werden. Andere Punktdichten und Größen können je nach Art des verwendeten Druckers angepaßt werden.

Tricks machen den Profi

Die bekannte Rubrik »Tips & Tricks« darf ebenfalls nicht vergessen werden. Sie fehlt natürlich auch in Sonderheft 10/86 nicht. Hier findet man beispielsweise einen verbesserten RENUMBER-Befehl, einen komfortablen Variablen-Dump sowie eine weitere Hardcopy-Routine, die nun allerdings den Inhalt des 80-Zeichen-Textbildschirms auf dem Drucker bringt.

Die vorgestellten Tips geben Ihnen gute Hilfen, Programmiererfahrung zu sammeln, die Sie schließlich Schritt für Schritt zum C128-Profi machen. Ohne Zweifel wird noch so mancher findige Kopf unter den C128-Besitzern und 64'er-Lesern dem C128 weitere Geheimnisse entlocken können, so daß das dritte Sonderheft, das Sie jetzt gerade lesen, bestimmt nicht das letzte sein wird.

(Michael Thomas/kn)

Checksummer 128 und MSE

Diese beiden Programme sind unentbehrlich beim Abtippen unserer Listings. Sie helfen Tippfehler zu vermeiden und sparen eine Menge Zeit.

Nobody is perfect. Jeder Computer-Fan, egal ob blutiger Anfänger oder ausgefuchster Profi, macht beim Abtippen von Programmen Tippfehler. Diese Fehler später zu finden, kann ein langwieriges Unterfangen werden.

Deshalb haben wir für Sie die Programme »Checksummer 128« und »MSE« (Maschinen-Sprache-Editor) entwickelt. Der Checksummer ist für Basic-Programme und der MSE für Maschinensprache-Listings zuständig.

Der MSE

Wie eben erwähnt, dient der MSE zur Eingabe von Maschinensprache-Programmen. Als erstes müssen Sie den sogenannten »MSE-Lader« (Listing 1) abtippen. Dieser erzeugt das eigentliche MSE-Programm auf Diskette oder Kassette.

Wichtig: Vor dem Eintippen des MSE-Laders müssen Sie unbedingt ein paar Befehle eingeben (ohne Basic-Zeilennummer): POKE 44,32 : POKE 8192,0 : NEW und <RETURN> drücken. Die Eingabe des MSE-Laders muß im C 64-Modus erfolgen, ebenso die Eingabe der MSE-Listings!

Jetzt können Sie beginnen, das Listing 1 abzutippen. Der MSE-Lader erkennt, wenn Sie beim Eintippen der DATA-Zeilen einen Fehler gemacht haben.

Wenn Sie das Listing 1 nicht auf einmal abtippen möchten, müssen Sie vor jedem neuen Laden des Programms unbedingt die oben genannte POKE-Zeile eingeben!

Datasetten-Besitzer müssen die »8« am Ende von Zeile 343 in eine »1« ändern.

Wenn Sie alles richtig gemacht haben und das Programm fehlerfrei abgetippt wurde, speichert es sich selbst auf Diskette oder Kassette unter dem Namen »MSE V1.1«. Dieses fertige MSE-Programm laden Sie dann bei Bedarf wie ein normales Basic-Programm und starten es mit »RUN«.

So arbeitet man mit dem MSE

Als erstes möchte der MSE den Namen des zu bearbeitenden Programms wissen. Dieser steht in der ersten Zeile unserer MSE-Listings. Dann müssen Sie die Start- und Endadresse des Programms eingeben. Dies sind die letzten beiden, vierstelligen Hexadezimalzahlen in der ersten Zeile unserer Listings.

Wenn Sie ein Programm von Diskette oder Kassette laden wollen, um an einer bestimmten Stelle weiterzutippen oder noch eine Korrektur vorzunehmen, geben Sie auf die Frage nach der Startadresse ein »L« ein. Danach müs-

```
5 PRINT CHR$(14)                                         <KD2>
10 PRINT "{CLR}"                                         <1HM>
20 PRINT "Q****************************W"                <G9D>
30 PRINT "{4DOWN,2SPACE}TEST{2SPACE,BLUE,6SPACE}"        <QGS>
40 PRINT "BBBBBBBBBBBBBBBBBBBBBBBBBBBB"                  <BAG>
```

Bild 1. Die Bedeutung der Steuerzeichen wird im nachfolgenden Text erklärt

In Zeile 10 müssen Sie nach den Anführungsstrichen die Tasten <SHIFT CLR/HOME> drücken und nicht die Klammern mit dem Wort CLR eingeben. In Zeile 20 drücken Sie nach den Anführungsstrichen die CBM-Taste und den Buchstaben <Q>, gefolgt von mehreren SHIFT- und Stern-Tasten und zum Schluß die CBM-Taste und den Buchstaben <W>. In Zeile 30 ist es viermal die CURSOR-abwärts-Taste, gefolgt von zweimaliger Leertaste, dann <SHIFT T> und normal EST, zum Schluß noch zweimal die Leertaste, die Farbtaste Blau <CTRL 7> und sechsmal die Leertaste. Zeile 40 besteht lediglich aus mehreren Grafikzeichen, die mit der CBM-Taste und <B> erzeugt werden.

sen Sie »D« oder »T« drücken, je nachdem, ob Sie von Diskette oder Kassette (»tape«) laden möchten. Wenn das Programm unter diesem Namen nicht auf der Diskette vorhanden ist oder ein sonstiger Ladefehler vorlag, meldet sich der MSE mit »I/O-ERROR«. In so einem Fall drücken Sie <RUN/STOP RESTORE> und geben einfach noch einmal »RUN« ein.

Beim Abtippen geben Sie hintereinander die abgedruckten Buchstaben und Zahlen des jeweiligen Listings ein. Für die Leerzeichen zwischen den Zahlen sorgt der MSE automatisch selbst. Wenn Sie in einer Zeile einen Tippfehler gemacht haben, meldet sich der MSE sofort mit einem Brummton und der Meldung »EINGABEFEHLER«. Nach einem Druck auf die RETURN-Taste können Sie mit der DEL-Taste den Fehler korrigieren.

Wenn Sie das gewünschte Programm vollständig eingegeben haben, speichert es der MSE automatisch auf Diskette oder Kassette.

Bei längeren Listings ist es unwahrscheinlich, daß Sie das komplette Programm auf einmal eingeben. Sie können Ihre bisherige Tipparbeit jederzeit durch <CTRL S> auf Diskette oder Kassette speichern und Ihr Werk später fortsetzen. Sie sollten sich dann allerdings im Heft markieren, wie weit Sie beim Abtippen gekommen sind! Später geben Sie dann nach dem Laden des ersten Programmteils <CTRL N> ein und auf die dann folgende Frage nach der Startadresse die Zeilennummer (Adresse), bei der Sie aufgehört haben zu tippen.

<CTRL M> erlaubt Ihnen jederzeit, Ihr Werk listen zu lassen. Durch <SPACE> können Sie weiterlisten lassen und durch <RUN/STOP> das Listen abbrechen.

Wenn Sie einen Drucker besitzen, können Sie das Programm auch mit <CTRL P> ausdrucken.

Mit <CTRL L> wird das Programm noch einmal neu in Ihren C 128 geladen.

Der Checksummer 128

Zuerst einmal müssen Sie das Checksummer-Programm (siehe Listing 2) mit dem MSE im C64-Modus abtippen. Die Eingabehinweise entnehmen Sie bitte obigen Angaben zur Eingabe von MSE-Listings. Wenn Sie fertig sind, speichern Sie das Programm auf Diskette oder Kassette.

Jetzt geht es los:

1. Starten Sie den Checksummer (im C128-Modus!) durch die Eingabe von »RUN« und das Drücken der RETURN-Taste.
2. Wenn die Meldung »Checksummer 128 aktiv « auf dem Bildschirm erscheint, haben Sie keinen Tippfehler gemacht und der Checksummer ist nun eingeschaltet.
3. Nun können Sie den Checksummer testen. Geben Sie bitte folgende Zeile ein und drücken Sie die RETURN-Taste:

10 REM

In der linken oberen Bildschirmecke sehen Sie nun die Prüfsumme der eben eingegebenen Basic-Zeile. Sie muß <A99> lauten. Dem Checksummer ist es übrigens egal,

Steuerzeichen	Bedeutung
{CTRL}	steht für Control-Taste, so bedeutet {CTRL+A}, daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:
{DOWN}	Taste neben rechtem Shift, Cursor unten
{UP}	Shift-Taste & Taste neben rechtem Shift; Cursor hoch
{CLR}	Shift-Taste & 2. Taste von ganz rechts oben
{INST}	Shift-Taste & Taste ganz rechts oben
{HOME}	2. Taste von ganz rechts oben
{DEL}	Taste ganz rechts oben
{RIGHT}	Taste ganz rechts unten
{LEFT}	Shift-Taste & Taste unten rechts
{SPACE}	Leertaste, Hinweis: {13 SPACE} bedeutet 13mal die Leertaste drücken
{SHIFT-SPACE}	Shift-Taste & Leertaste
{F1}	grauer Tastenblock rechts oben
{F3}	grauer Tastenblock rechts oben
{F5}	grauer Tastenblock rechts oben
{F7}	grauer Tastenblock rechts oben
{F2}	grauer Tastenblock rechts oben & Shift
{F4}	grauer Tastenblock rechts oben & Shift
{F6}	grauer Tastenblock rechts oben & Shift
{F8}	grauer Tastenblock rechts oben & Shift
{RETURN}	Return-Taste
{CTRL+I}	TAB-Taste
{CTRL+J}	Line-Feed-Taste

Steuerzeichen	Bedeutung
{CTRL}	Control-Taste
{BLACK}	Control-Taste & 1
{WHITE}	Control-Taste & 2
{RED}	Control-Taste & 3
{CYAN}	Control-Taste & 4
{PURPLE}	Control-Taste & 5
{GREEN}	Control-Taste & 6
{BLUE}	Control-Taste & 7
{YELLOW}	Control-Taste & 8
{RVSON}	Control-Taste & 9
{RVOFF}	Control-Taste & 0
{ORANGE}	Commodore-Taste & 1
{BROWN}	Commodore-Taste & 2
{LIG.RED}	Commodore-Taste & 3
{GREY 1}	Commodore-Taste & 4
{GREY 2}	Commodore-Taste & 5
{LIG.GREEN}	Commodore-Taste & 6
{LIG.BLUE}	Commodore-Taste & 7
{GREY 3}	Commodore-Taste & 8

Wenn Sie sich erst einmal an die in Klartext geschriebenen Steuerzeichen gewöhnt haben, werden Sie den Vorteil dieser Schreibweise erkennen. Der zu dem jeweiligen Steuerzeichen gehörende Klartext ist so verfaßt, daß Sie leicht die Taste beziehungsweise die Tastenkombination finden, die Sie drücken müssen.

Tabelle 1. Die Steuerbefehle für den C128 im Klartext

ob Sie »10 REM« oder »10REM« eintippen. Nur innerhalb von Anführungszeichen ist die richtige Anzahl an Leerzeichen wichtig. Diese Prüfsummen erscheinen (sofern Sie den Checksummer 128 eingeschaltet haben) immer dann, wenn Sie eine Basic-Zeile eintippen und dann die RETURN-Taste drücken. In diesem Sonderheft finden Sie die Prüfsumme immer am Ende jeder Programmzeile.

Diese Zahlen dürfen Sie NICHT mit abtippen. Sie dienen lediglich zur Kontrolle, ob Sie alles richtig eingegeben haben.

Als Beispiel können Sie Bild 1 betrachten. Am rechten Rand jeder Spalte sehen Sie die Prüfsummen in spitzen Klammern.

Damit sind wir beim zweiten wichtigen Punkt: Sehen Sie sich die Zeile 343 von Listing 1 genauer an. Nach dem ersten Anführungszeichen nach dem PRINT-Befehl sehen Sie ein Zeichen, das Sie auf der Tastatur des C 128 vergeblich suchen werden: die geschweifte Klammer »{«. Immer, wenn Sie in einem unserer Listings diese Klammern sehen, dürfen Sie das, was innerhalb der Klammern steht, nicht eintippen. Sie müssen die entsprechende Taste drücken. Beispiel:

10 PRINT "{CLR}"

bedeutet: Nach dem Anführungszeichen die »Bildschirmlöschen«-Taste drücken (< SHIFT CLR/HOME >). In Tabelle 1 sehen Sie eine Zusammenfassung aller möglichen Steuertasten und des entsprechenden Klartextes.

Wenn Sie Bild 1 betrachten, sehen Sie in Zeile 30 innerhalb der Anführungszeichen ein unterstrichenes »T« nach der geschweiften Klammer. Das bedeutet, daß Sie ein »T« zusammen mit der SHIFT-Taste drücken müssen, also < SHIFT T >. Wenn ein Zeichen »überstrichen« ist, müssen Sie dieses zusammen mit der CBM-Taste eingeben. Die CBM-Taste befindet sich ganz links unten auf der Tastatur und hat die Aufschrift »C=«. Auf dem Bildschirm sehen Sie die entsprechenden Grafikzeichen (siehe Handbuch zum Computer, im Anhang).

(G. Möllmann, N. Mann, D. Weineck, A. Lietz)

```
100 REM DIESES PROGRAMM ERZEUGT DEN
110 REM MSE V1.1 AUF DISKETTE.
120 REM BESITZER EINER DATASETTE
130 REM MUESSEN DIE '8' AM ENDE VON
140 REM ZEILE 343 IN EINE '1' AENDERN!
150 REM
230 IF PEEK(44)<>32 THEN PRINT "{CLR}SIE HABEN VE
    RGESSEN, DIE POKES EINZUGE- BEN!": END
240 PRINT "{CLR}";: DIM H(75): FOR I=0 TO 9
250 H(48+I)=I: H(65+I)=I+10: NEXT : Z=1000
260 FOR I=2048 TO 3755 STEP 20: PRINT "{HOME}ICH
    LESE ZEILE:"Z
261 FOR N=0 TO 19: READ A$: IF LEN(A$)<>2 THEN 90
    0
262 IF PEEK(63)+PEEK(64)*256<>Z THEN 800
270 H=ASC(LEFT$(A$,1)): L=ASC(RIGHT$(A$,1))
280 D=H(H)*16+H(L): S=S+D: POKE I+N,D
290 NEXT : READ V: IF S<>V THEN 900
300 S=0: Z=Z+1: NEXT : R=PEEK(2111): H=PEEK(2106)
301 POKE 53280,R: POKE 53281,H: POKE 646,R: PRINT
     "{CLR}DIE DATA-ZEILEN SIND FEHLERFREI!"
302 PRINT "SIE KOENNEN NUN DIE FARBEN DES MSE"
303 PRINT "EINSTELLEN.": PRINT "{2DOWN} {RVSON}DR
    UECKEN SIE <1>, <2> ODER <9>
304 PRINT "{DOWN,2SPACE}<1> - RAHMEN-/SCHRIFTFARB
    E
305 PRINT "{2SPACE}<2> - HINTERGRUNDFARBE
306 PRINT "{DOWN,2SPACE}<9> - FARBEN UEBERNEHMEN
307 PRINT "{2DOWN}FARBE <1>:"R: PRINT "FARBE <2>:
    "H
308 GET A: IF A=0 THEN 308
309 IF A=1 THEN R=(R+1) AND 15
310 IF A=2 THEN H=(H+1) AND 15
311 IF A=9 THEN 340
312 GOTO 301
340 POKE 2106,H: POKE 2111,R
342 POKE 631,19: POKE 632,13: POKE 198,2
343 PRINT "{CLR}SAVE" CHR$(34)"MSE V1.1" CHR$(34)
    ",8
344 POKE 43,1: POKE 44,8: POKE 45,172: POKE 46,14
    : END
800 PRINT "{CLR,RVSON}SIE HABEN ZEILE"Z"{LEFT} VE
    RGESSEN:": A=PEEK(646) AND 15
810 POKE 646,PEEK(53281) AND 15: PRINT "LIST"Z-2"
    -"Z+2: POKE 646,A
820 GOTO 920
900 PRINT "{CLR,RVSON}SIE HABEN EINEN TIPPFEHLER
    GEMACHT:": A=PEEK(646) AND 15
910 POKE 646,PEEK(53281) AND 15: PRINT "LIST"Z: P
    OKE 646,A
920 POKE 631,19: POKE 632,17: POKE 633,13: POKE 1
    98,3: END
1000 DATA 00,0B,08,0A,00,9E,32,30,36,31,00,00,00,
     A2,08,A9,36,85,A4,A9, 1247
1001 DATA 08,85,A5,A9,00,85,A6,A9,B0,85,A7,A0,00,
     B1,A4,91,A6,C8,D0,F9, 2888
1002 DATA E6,A5,E6,A7,CA,D0,F2,A9,36,85,01,4C,00,
     B0,20,D1,B1,A9,00,8D, 2781
1003 DATA 21,D0,A9,0F,8D,20,D0,8D,86,02,A0,B3,A9,
     74,20,FF,B1,A0,B3,A9, 2679
1004 DATA B9,20,FF,B1,A0,00,20,CF,FF,99,01,02,C8,
     C9,0D,D0,F5,88,F0,D2, 2912
1005 DATA C0,11,90,02,A0,10,8C,00,02,20,EA,B1,A0,
     B3,A9,CF,20,FF,B1,20, 2327
1006 DATA 8E,B4,85,FC,85,62,20,8E,B4,85,FB,85,61,
     20,A7,B4,D0,20,A0,B3, 2864
1007 DATA A9,E5,20,FF,B1,20,8E,B4,85,60,20,8E,B4,
     85,5F,20,A7,B4,D0,0A, 2624
1008 DATA A5,61,C5,5F,A5,62,E5,60,90,06,20,43,B3,
     4C,3A,B0,A9,AA,A0,00, 2379
1009 DATA EA,EA,E6,FB,D0,02,E6,FC,20,3F,B2,90,EF,
     4C,FB,B4,A2,02,86,58, 3190
1010 DATA A9,A6,A0,9D,20,F2,B1,20,E4,FF,F0,FB,C9,
     30,90,0C,C9,47,B0,08, 2970
1011 DATA C9,3A,90,0B,C9,41,B0,07,C9,14,D0,0F,4C,
     0B,B1,20,D2,FF,A6,58, 2322
1012 DATA 95,F7,C6,58,D0,D2,60,AE,8D,02,F0,26,C9,
     0C,D0,03,4C,0B,B6,C9, 2685
1013 DATA 13,D0,03,4C,8B,B5,C9,0D,D0,03,4C,BA,B4,
     C9,10,D0,03,4C,68,B5, 2282
1014 DATA C9,0E,D0,06,20,5F,B4,4C,64,B1,4C,92,B0,
     A5,F9,20,02,B1,0A,0A, 2132
1015 DATA 0A,0A,85,F9,A5,F8,20,02,B1,05,F9,60,C9,
     3A,90,02,69,08,29,0F, 1950
1016 DATA 60,A6,59,E0,08,90,1F,A6,58,E0,02,B0,06,
     20,D2,FF,4C,8E,B0,C6, 2509
1017 DATA 59,A0,14,A9,92,20,F2,B1,CA,D0,FA,84,57,
     68,68,4C,8B,B1,A6,D3, 2891
1018 DATA E0,08,B0,03,4C,92,B0,20,D2,FF,A6,58,E0,
     02,90,09,C6,59,20,D2, 2468
1019 DATA FF,C6,58,D0,F9,4C,8E,B0,48,4A,4A,4A,4A,
     20,59,B1,68,29,0F,C9, 2419
1020 DATA 0A,90,02,69,06,69,30,4C,D2,FF,A2,FC,9A,
     20,D1,B1,20,48,B2,20, 2261
1021 DATA EA,B1,20,9F,B2,A5,FC,20,4E,B1,A5,FB,20,
     4E,B1,20,ED,B1,A9,3A, 2860
1022 DATA A0,20,20,F2,B1,A9,00,85,59,20,8E,B0,20,
     ED,B1,A4,59,20,EF,B0, 2530
1023 DATA 91,FB,C8,84,59,C0,08,90,EC,20,10,B2,A9,
     12,20,D2,FF,20,8E,B0, 2657
1024 DATA 20,EF,B0,C5,FF,F0,0D,20,43,B3,A9,14,A0,
     14,20,F2,B1,4C,A2,B1, 2665
1025 DATA A9,92,20,D2,FF,20,33,B2,20,E0,B2,20,3F,
     B2,90,9F,4C,8B,B5,A9, 2648
1026 DATA 93,20,D2,FF,A2,00,A9,03,9D,00,D8,9D,00,
     D9,9D,00,DA,9D,00,DB, 2476
1027 DATA E8,D0,EF,60,A9,0D,2C,A9,20,4C,D2,FF,20,
     D2,FF,98,4C,D2,FF,20, 2965
1028 DATA E4,FF,F0,FB,60,84,5D,85,5C,A0,00,B1,5C,
     F0,06,20,D2,FF,C8,D0, 3100
1029 DATA F6,60,A5,FB,85,5A,A0,00,84,5B,B1,FB,18,
     65,5A,85,5A,90,02,E6, 2606
1030 DATA 5B,06,5A,26,5B,C8,C0,08,90,EC,A5,5A,65,
     5B,85,FF,60,18,A5,FB, 2467
1031 DATA 69,08,85,FB,90,02,E6,FC,60,A5,FB,C5,5F,
     A5,FC,E5,60,60,A0,B3, 3106
1032 DATA A9,FB,20,FF,B1,A0,01,B9,00,02,20,D2,FF,
     CC,00,02,C8,90,F4,A9, 2692
```

```
1033 DATA 14,ED,00,02,AA,20,ED,B1,CA,D0,FA,A5,62,
     20,4E,B1,A5,61,20,4E, 2457
1034 DATA B1,20,ED,B1,A5,60,20,4E,B1,A5,5F,20,4E,
     B1,EA,EA,EA,EA,EA,EA, 3122
1035 DATA EA,EA,24,5E,10,01,60,A9,12,20,D2,FF,A2,
     28,20,ED,B1,CA,D0,FA, 2703
1036 DATA A9,92,4C,D2,FF,A5,D6,C9,16,B0,01,60,A9,
     A0,85,A4,A9,78,85,A6, 2945
1037 DATA A9,04,85,A5,85,A7,A2,13,A0,27,B1,A4,91,
     A6,88,10,F9,CA,F0,19, 2671
1038 DATA 18,A5,A4,69,28,85,A4,90,02,E6,A5,18,A5,
     A6,69,28,85,A6,90,E0, 2503
1039 DATA E6,A7,4C,B6,B2,A9,91,4C,D2,FF,A9,0F,8D,
     18,D4,A9,00,8D,05,D4, 2776
1040 DATA A9,F7,8D,06,D4,A9,11,8D,04,D4,A9,32,8D,
     01,D4,A9,00,8D,00,D4, 2413
1041 DATA A0,80,20,09,B3,A9,10,8D,04,D4,60,A2,FF,
     CA,D0,FD,88,D0,F8,60, 2914
1042 DATA A9,0F,8D,18,D4,A9,2D,8D,05,D4,A9,A5,8D,
     06,D4,A9,21,8D,04,D4, 2385
1043 DATA A9,07,8D,01,D4,A9,05,8D,00,D4,A0,FF,20,
     09,B3,A9,20,8D,04,D4, 2250
1044 DATA A9,00,8D,01,D4,8D,00,D4,60,38,20,F0,FF,
     8A,48,98,48,18,A0,06, 2179
1045 DATA A2,18,20,F0,FF,A0,B4,A9,0A,20,FF,B1,20,
     12,B3,20,E4,FF,F0,FB, 2931
1046 DATA A2,1D,A9,14,20,D2,FF,CA,D0,FA,68,A8,68,
     AA,18,4C,F0,FF,0D,0D, 2704
1047 DATA 0D,20,20,20,20,20,20,20,4D,41,53,43,48,
     49,4E,45,4E,53,50,52, 1144
1048 DATA 41,43,48,45,20,2D,20,45,44,49,54,4F,52,
     20,0D,0D,20,20,20,20, 1023
1049 DATA 20,20,20,20,56,4F,4E,20,4E,2E,4D,41,4E,
     4E,20,26,20,44,2E,57, 1128
1050 DATA 45,49,4E,45,43,4B,00,0D,0D,0D,20,20,20,
     50,52,4F,47,52,41,4D, 1102
1051 DATA 4D,4E,41,4D,45,20,3A,20,00,0D,0D,20,20,
     20,53,54,41,52,54,41, 1073
1052 DATA 44,52,45,53,53,45,20,3A,20,24,00,0D,0D,
     20,20,20,45,4E,44,41, 1014
1053 DATA 44,52,45,53,53,45,20,20,20,3A,20,24,00,
     92,01,01,50,52,4F,47, 1136
1054 DATA 52,41,4D,4D,20,3A,20,00,12,20,20,2A,2A,
     2A,20,46,41,4C,53,43, 1024
1055 DATA 48,45,20,45,49,4E,47,41,42,45,20,2A,2A,
     2A,20,20,92,00,0D,0D, 1058
1056 DATA 2A,2A,2A,20,45,4E,44,45,20,2A,2A,2A,00,
     13,01,20,20,12,44,92, 916
1057 DATA 49,53,4B,20,4F,44,45,52,20,12,54,92,41,
     50,45,0D,00,13,20,20, 1151
1058 DATA 49,2F,4F,20,2D,20,46,45,48,4C,45,52,00,
     20,D1,B1,20,48,B2,A0, 1606
1059 DATA B3,A9,CF,20,FF,B1,20,8E,B4,85,FC,20,8E,
     B4,85,FB,C5,61,A5,FC, 3207
1060 DATA E5,62,90,23,A5,FB,C5,5F,A5,FC,E5,60,B0,
     19,20,A7,B4,D0,14,60, 2860
1061 DATA 20,A7,B4,F0,0C,85,F9,20,A7,B4,F0,05,85,
     F8,4C,EF,B0,68,68,20, 2749
1062 DATA 43,B3,4C,5F,B4,20,CF,FF,C9,4C,D0,09,20,
     D1,B1,20,48,B2,4C,0B, 2372
1063 DATA B6,C9,0D,60,A9,00,85,5E,20,5F,B4,20,EA,
     B1,20,0D,B5,24,5E,30, 2042
1064 DATA 05,20,E4,FF,F0,FB,20,E1,FF,F0,26,20,9F,
     B2,24,5E,10,09,20,4E, 2435
1065 DATA B5,20,0D,B5,20,60,B5,20,33,B2,20,3F,B2,
     90,D7,A0,B4,A9,28,20, 2190
1066 DATA FF,B1,20,E4,FF,C9,0D,D0,F9,A9,00,85,5E,
     A5,61,85,FB,A5,62,85, 3056
1067 DATA FC,20,E0,B2,4C,64,B1,A5,FC,20,4E,B1,A5,
     FB,85,FF,20,4E,B1,A9, 3003
1068 DATA 20,A0,3A,20,F2,B1,A0,00,20,ED,B1,B1,FB,
     20,4E,B1,C8,C0,08,90, 2566
1069 DATA F3,20,ED,B1,24,5E,30,03,A9,12,2C,A9,20,
     20,D2,FF,20,10,B2,A5, 2190
1070 DATA FF,20,4E,B1,A9,92,20,D2,FF,4C,EA,B1,A9,
     FF,85,B8,85,B9,A9,04, 3073
1071 DATA 85,BA,20,C0,FF,A2,FF,4C,C9,FF,20,CC,FF,
     A9,FF,4C,C3,FF,20,5F, 3315
1072 DATA B4,A9,80,85,5E,20,4E,B5,20,48,B2,A2,24,
     A9,2D,20,D2,FF,CA,D0, 2596
1073 DATA FA,20,EA,B1,20,EA,B1,20,60,B5,4C,C1,B4,
     20,B8,B5,A6,5F,A4,60, 2812
1074 DATA A9,61,20,D8,FF,B0,0A,20,B7,FF,29,BF,D0,
     03,4C,FB,B4,A9,01,20, 2577
1075 DATA C3,FF,20,68,B6,A0,B4,A9,4F,20,FF,B1,20,
     F9,B1,4C,FB,B4,20,68, 2921
1076 DATA B6,A9,37,A0,B4,20,FF,B1,20,F9,B1,A2,08,
     C9,44,F0,06,A2,01,C9, 2717
1077 DATA 54,D0,F1,A9,01,A8,20,BA,FF,A0,00,E0,01,
     F0,1A,A9,40,8D,20,02, 2403
1078 DATA A9,3A,8D,21,02,B9,01,02,99,22,02,C8,CC,
     00,02,90,F4,C8,C8,D0, 2182
1079 DATA 0C,B9,01,02,99,20,02,C8,CC,00,02,D0,F4,
     98,A2,20,A0,02,4C,BD, 2018
1080 DATA FF,20,B8,B5,A5,BA,C9,08,90,33,A6,B9,86,
     57,A9,01,20,C3,FF,A9, 2800
1081 DATA 60,85,B9,20,C0,FF,B0,28,A5,BA,20,B4,FF,
     A5,B9,20,96,FF,20,A5, 2911
1082 DATA FF,85,61,A5,90,4A,4A,B0,13,20,A5,FF,85,
     62,20,AB,FF,A5,57,85, 2663
1083 DATA B9,A9,00,20,D5,FF,90,03,4C,A3,B5,86,5F,
     84,60,A5,BA,C9,01,D0, 2639
1084 DATA 0A,AD,3D,03,85,61,AD,3E,03,85,62,4C,FB,
     B4,A9,13,20,D2,FF,A2, 2300
1085 DATA 1C,20,ED,B1,CA,D0,FA,60,00,00,00,00,00,
     00,00,00,00,00,00,00, 1230
```

Listing 1. Der »MSE«-Lader für die Eingabe von Maschinen-Programmen

```
Name : checksummer 128    1c01 1e9b
-----------------------------------
1c01 : 29 1c c3 07 fe 02 30 3a  3f
1c09 : 99 22 93 43 48 45 43 4b  53
1c11 : 53 55 4d 4d 45 52 20 31  d6
1c19 : 32 38 20 41 4b 54 49 56  c1
1c21 : 22 3a 9e 37 32 31 31 00  60
1c29 : 00 00 a9 80 85 9d a9 00  90
1c31 : 85 7f a9 ff 85 3c a9 59  74
1c39 : 85 4e a9 1c 85 4f a9 22  91
1c41 : 85 50 a9 0b 85 51 a0 00  1f
1c49 : a2 02 b1 4e 91 50 c8 d0  83
1c51 : f9 e6 4f e6 51 ca 10 f2  00
1c59 : 4c 2a 0b 00 00 00 00 00  7d
1c61 : a9 00 8d 00 ff a9 0d 20  2f
1c69 : d2 ff ad cf 41 c9 56 d0  fd
1c71 : 11 ad d0 41 c9 37 d0 0a  63
1c79 : ad d1 41 c9 2e d0 03 a9  61
1c81 : ff 2c a9 00 8d 25 0b aa  84
1c89 : d0 0b 8d 64 0d a9 65 85  8d
1c91 : 2b a9 0d 85 2c 20 cd 0b  96
1c99 : a9 6c 8d 02 03 a9 0b 8d  e1
1ca1 : 03 03 2c 27 0b 10 06 ee  3d
1ca9 : 27 0b 4c 35 0d 2c 26 0b  f0
1cb1 : 10 03 4c f9 0c 20 a4 0b  00
1cb9 : 20 af 0b 20 c2 0b aa f0  89
1cc1 : e1 b0 0c ce 26 0b 20 e3  da
1cc9 : 0b 20 af 0b 20 c2 0b 2c  ce
1cd1 : 25 0b 30 03 4c 94 a4 4c  7d
1cd9 : d7 4d 2c 25 0b 30 03 4c  dd
1ce1 : 60 a5 4c 93 4f a9 ff a2  21
1ce9 : 01 2c 25 0b 30 05 85 7a  e1
1cf1 : 86 7b 60 85 3d 86 3e 60  bf
1cf9 : 2c 25 0b 30 03 4c 73 00  e1
1d01 : 4c 80 03 2c 25 0b 30 03  45
1d09 : 4c 44 a6 4c d9 51 2c 25  ce
1d11 : 0b 30 03 4c 13 a6 4c 64  df
1d19 : 50 2c 25 0b 30 06 20 6b  b5
1d21 : a9 4c f1 0b 20 a0 50 a2  5c
1d29 : 00 2c 25 0b 10 02 a2 02  8a
1d31 : b5 15 85 aa b5 14 85 a9  8d
1d39 : 60 a9 00 85 a7 85 a8 a9  bb
1d41 : ff 85 fc a9 07 85 fd 20  4c
1d49 : 9d 0c f0 77 c9 20 f0 06  85
1d51 : 20 8e 0c 4c 11 0c 20 9d  72
1d59 : 0c f0 68 c9 20 f0 f7 c9  2e
1d61 : 3a d0 06 20 8e 0c 4c 20  44
1d69 : 0c c9 52 d0 29 20 9d 0c  2b
1d71 : c9 45 d0 1a 20 9d 0c c9  07
1d79 : 4d d0 0b 20 9d 0c f0 43  7a
1d81 : c9 3a d0 f7 f0 dd a5 fc  29
1d89 : d0 02 c6 fd c6 fc a5 fc  b0
1d91 : d0 02 c6 fd c6 fc a5 fc  b8
1d99 : d0 02 c6 fd c6 fc 20 9d  ec
1da1 : 0c f0 20 c9 20 f0 f7 c9  64
1da9 : 3a f0 b8 c9 22 f0 06 20  c5
1db1 : 8e 0c 4c 68 0c 20 8e 0c  7a
1db9 : 20 9d 0c f0 06 c9 22 d0  a2
1dc1 : f4 f0 ec 60 06 a7 26 a8  fc
1dc9 : 08 46 a7 28 26 a7 45 a7  e7
1dd1 : 85 a7 60 e6 fc d0 02 e6  4b
1dd9 : fd a0 00 b1 fc 60 ea ea  b1
1de1 : ea ea ea ea ea ea ea ea  e0
1de9 : ea a5 a8 45 a7 85 a8 20  02
1df1 : bf 0c 20 bf 0c a9 00 a2  0a
1df9 : 04 06 a7 26 a8 2a ca 10  d6
1e01 : f8 c9 0a 90 02 69 06 69  c9
1e09 : 30 4c d2 ff a9 00 8d 00  e5
1e11 : ff 38 20 f0 ff 8a 48 a9  1b
1e19 : 13 20 d2 ff a9 3c 20 d2  93
1e21 : ff 20 b3 0c a9 3e 20 d2  51
1e29 : ff 68 aa 18 4c f0 ff ad  b2
1e31 : 27 03 48 ad 26 03 48 a9  91
1e39 : 49 8d 26 03 a9 0d 8d 27  ba
1e41 : 03 2c 25 0b 10 03 a2 02  ad
1e49 : 2c a2 00 a5 a9 95 14 a5  5e
1e51 : aa 95 15 20 d8 0b a9 00  9c
1e59 : 8d 28 0b 2c 25 0b 30 08  be
1e61 : ce 27 0b a0 02 4c d8 a6  cd
1e69 : 20 f8 50 20 03 0c ee 26  b6
1e71 : 0b 68 8d 26 03 68 8d 27  d0
1e79 : 03 20 d6 0c 4c 6c 0b 85  23
1e81 : ab 8a 48 ae 28 0b a5 ab  22
1e89 : c9 0d d0 02 a9 00 9d 00  5e
1e91 : 08 ee 28 0b 68 aa a5 ab  46
1e99 : 18 60 28 4a 2f 4e 29 20  7f
```

Listing 2. Der »Checksummer 128« für die Eingabe von Basic-Listings

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Geschäftsführender Chefredakteur: Michael Scharfenberger

Chefredakteur: Albert Absmeier
Stellv. Chefredakteur: Georg Klinge
Leitender Redakteur: Gottfried Knechtel (kn)
Redaktion: Klaus Schrödl (sk), Boris Schneider (bs)

Mitarbeiter der Redaktion: Martin Jobst, Andreas Lietz, Florian Müller, Axel Pretzsch, Michael Thomas, Dr. Rudolf Egg

Layout: Leo Eder (Leitung), Rolf Raß (Cheflayouter) Andrea Miller, Katja Milles

Fotografie: Jens Jancke

Titelgestaltung: Andrea Miller

Produktionsleiter: Klaus Buck

Anzeigenverkaufsleitung: Ralph-Peter Rauchfuss

Anzeigenverkauf: Britta Fiebig (282)

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-41 56 56, Telex: 862329
USA: M&T Publishing Inc.; 501 Galveston Drive Redwood City, CA 94063 Telefon: (415) 366-3600

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen und dazu, daß Markt&Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Marketingleiter: Hans Hörl (114)

Vertriebsleiter: Helmut Grünfeldt (189)

Anzeigenverwaltung und Disposition: Lisa Landthaler (233)

Druck: SOV St. Otto-Verlag GmbH, Laubanger 23, 8600 Bamberg

Bezugsmöglichkeiten: Leser-Service: Telefon (089) 4613-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Preis: Das Einzelheft kostet DM 14,–

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Straße 96, 7000 Stuttgart 1, Telefon (0711) 6483-0

Verantwortlich:
Für redaktionellen Teil: Albert Absmeier
Für Anzeigen: Britta Fiebig

Redaktionsdirektor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0, Telex 5-22052

ISSN 0931-8933





